APPROVAL OF ACCEPTANCE

A project report written by JANNATUL FERDOUSE MEEM (16502000082), NAHID MAHMUD SHUVA (16502000105), HASNAIN MUHAMMAD HASIB (16502000095) entitled “Autonomous UGV using SLAM and Computer Vision” submitted to the, Department of Computer Science and Engineering, Barisal Information Technology College (BITC) for partial fulfilment of the requirements for the degree of B.Sc.(Hon’s) in Computer Science and Engineering. The project is done under the supervision of JAKARIA ISLAM EMON Lecturer BARISAL INFORMATION TECHNOLOGY COLLEGE.

1st Examiner

2nd Examiner

Accepted by:

Jakaria Islam Emon

Lecturer

Barisal Information Technology College

DECLARATION

We hereby declare that this report has been done by JANNATUL FERDOUSE MEEM, NAHID MAHMUD SHUVA, HASNAIN MUHAMMAD HASIB under the supervision of JAKARIA ISLAM EMON. We also declare that all the information furnished in this capstone project report is based on our intensive research and is genuine.

This capstone does not, to the base of our knowledge, contain part of our work which has been submitted for the award of our degree either of this university or any university without proper citation.

Accepted by:

Jakaria Islam Emon

Lecturer

Barisal Information Technology College (BITC)

Submitted By:

JANNATUL FERDOUSE MEEM

Reg. No: 16502000082

NAHID MAHMUD SHUVA

Reg. No:16502000105

HASNAIN MUHAMMAD HASIB

Reg. No:16502000095

ACKNOWLEDGEMENT

This capstone project has taken a lot of time and work. It would not have been feasible, though, without the help of my classmates and teachers. We would like to express our heartfelt gratitude to each and every one of them. We are really grateful to Jakaria Islam Emon lecturer of the Department of CSE at BITC, for his unwavering support, direction, and constant monitoring, as well as for giving vital project information and for their assistance in completing our research. We are grateful to Anisur Rahman Sir, the Principal of BITC for allowing us to use the department's facilities to complete the project effectively. We'd want to show our gratitude.

PREFACE

It is virtually hard to keep up with the evolving trends in today's situation, where the area of Computer Science and Engineering has revolutionized and where breakthroughs happen in the blink of an eye. Excellence is achieved when we care more than others believe is prudent, take more risks than others believe is prudent, dream bigger than others believe is feasible, and demand more than others believe is achievable. That mindset is what leads to numerous inventions as well as new adjustments to current technologies. A well-thought-out and well-executed initiative undertaken with the latter mindset will enable the world to feel the power of knowledge. A team of three students with complementary skills committed to a common idea of the project have applied their approach for which we hold ourselves completely accountable. During this project, the aforementioned team gets the real, firsthand experience for working in an actual environment of communicating electronics. The majority of their theoretical knowledge obtained during their education is put to the test here. We got the chance to gain real-world experience, which greatly expanded our knowledge base. Many people believe that leadership cannot be taught, which is accurate but unimportant, as our supervisor, Jakaria Islam Emon Sir, demonstrates. His guidance, along with his skills, has steered our project in a positive direction. As a result, we were entrusted with a real-life project, the completion of which had eventually led us into the field of communication's continuing innovations and revolution.

ABSTRACT

A smart Unmanned Ground Vehicle (UGV) is designed and developed for some application specific missions to operate in different environments Since autonomous navigation and urban search and rescue is a difficult domain for autonomous mobile drivers to operate in. In our work, we have designed a small rover to operate Autonomously in different terrains with or without daylight. The UGV can send visual feedback to the operator at a remote location. It sends continuous ultrasonic signals data and video feed, environmental sensory data of the disaster site to the control station. It can run through rough terrain in a completely dark environment with the help of a powerful remote-controlled flashlight with Onboard LiDAR sensor camera can detect the obstacles around the UGV and can operate autonomously using SLAM technology, detect living humans inside the rubble and can establish communication between them and the rescuers. For a rover, terrain can be highly unstructured, with many obstacles and hazardous environments to deal with. Besides, if human rescue teams are going to accept rover assistance, they need to be assured that the rovers are going to be helpful, not a hindrance. The platform has been developed to be low in cost and easy to construct from commonly available parts so that they are suitable for developing countries like Bangladesh.

List of Figures

Contents

Abbreviations

UGV	Unmanned Ground Vehicle
RoS	Robotic Operating System
LiDAR	Light Detection and Ranging
SLAM	Simultaneous Localization and mapping
CNN	Convolutional Neural Network
YOLO	You only look once
GNSS	Global Navigation Satellite System
GPS	Global Positioning System
DC	Direct Current
IR	Infrared
DHT	Temperature and Humidity Sensor
RTH	Return To Home
FLCP	Fast Local Cylindrical Planning
IVG	Verification of Global planning
ACO	Ant Colony Optimization
SA	Simulated Annealing
FOI	Feature Of interest
USSR	Union Of Soviet Socialist Republics
SC	Strategic Computing
RSTA	Reconnaissance Surveillance Target Acquisition
NOSC	Naval Ocean System Centre
ATT	Advance Teleoperator Technology
PC	Personal Computer
ANN	Artificial Neural Network
AI	Artificial Intelligence
NLP	Natural Language Processing

RCCN	Region based Convolutional Neural Network
SOTA	State Of The Art
CUPR	Computer Vision and Pattern Recognition
FPS	Frame Per Second
IOU	Intersection Over Union
FOV	Field Of View
DoG	Difference of Gaussian
BRIEF	Binary Robust Independent Elementary Features
FLANN	Fast Library for Approximate Nearest Neighbors
RANSAC	Random sample consensus
PnP	Perspective n-Point
ICP	Iterative Closest Point
EKF	Extended Kalman Filter
IMU	Inercial Measurement Device
AHRS	Attitude and Heading Reference System
INS	Internal Navigation System
BoF	Bag of Features
BOVW	Bag Of Visual Wonde
HMI	Human Machine Interaction
NIMA	National Imagery Mapping Agency
LIPO	Lithium Polymer
FC	Flight Controller
VCC	Power
GND	Ground
LED	Light Emitting Diode
PWM	Pulse Width Modulation

Chapter 1

Introduction

1.1 Introduction:

Robotics is an important field of interest in modern age of automation. Unlike human being a computer controlled robot can work with speed and accuracy without feeling exhausted. A robot can also perform preassigned tasks in a hazardous environment, reducing the risks and threats of human beings. Unmanned Ground Vehicles (UGVs) are a type of mobile robots, which have numerous applications both in military and civilian domains. In the broad dictionary sense, an unmanned ground vehicle is any piece of mechanized equipment that moves across the surface of the ground and serves as a means of carrying or transporting something, but explicitly does not carry human being. The unmanned ground vehicles (UGVs) have shown applications in different areas including, military missions such as explosive ordnance disposal, reconnaissance, surveillance, search and rescue in disastrous areas, harvesting, transporting, patrol monitoring and detection, investigation, exploration and inspection at tunnels, buildings, etc. UGVs may rely on Sensors such as cameras, Optical Lidar, Sonar sensors, GPS, etc. In this paper, the proposed UGV can perform remote operations and some autonomous operations. This UGV rover can survive, monitor and 3d map the place, detect weapon and objects, collect and send environmental sensory information to the operator and also can operate some primary missions by remote control. UGVs can be equipped with a variety of sensors and payloads. Due to operating in indoor and other GNSS-denied environments, UGVs may rely on LiDAR sensors, combined with inertial navigation systems and vehicle odometer, for accurate navigation. Rovers are particularly suitable for demanding problems which require accurate low-speed maneuver and driving capabilities such as detailed area mapping. UGV with LiDAR can be used to examine the surface of the earth, assess information about the ground surface, create a digital twin of an object or detail a range of geospatial information. LiDAR systems harness this technology, using LiDAR data to map three-dimensional models and digital elevation. With this environmental sensory data they can effectively deploy rescue operations in specific regions of the whole area or in the specific building. When a disaster strikes, such as an earthquake, hurricane, typhoon, flood, dam failure, technological accident, fire, collapsed building due to natural or man-made events like terrorist activities or the release of hazardous materials, rescue workers have to step in, in order to prevent, or at least minimize, the loss of human lives. While the situation can vary a lot, the main objective of the rescue workers is to find humans in need of assistance (Search) and to move them out of harm’s way (Rescue). As this project’s main functionality is to deal with tough situations where human beings cannot handle situations like darkness, entering narrow and small places, exploring war and radioactive area and as we want to provide primary life support of victims in any disaster and have to keep the rescue team or special operation team safe, the best solution at hand is a compromise where humans and robots team-up. Where the robot does repetitive, dull, hazardous and dangerous parts of the work, while the human concentrates on finding patterns, making conclusions and helping the robot to understand what part of a task to concentrate on and in what order to execute its sub-goals. The general idea is to let the robot do things it does better than the human, and vice versa. Rovers designed to move on the ground and can detect objects, weapons, characterize them and send feedback to the remote station wirelessly without any human interference. In a war or hostage situation, it is difficult to detect whether a person holding a stick or a gun. To get more extensive, accurate, and real-time perception of the objects, weapons, and environment a Vision System is added to the Rover.

A certain level of autonomous flight capability is required for the vehicle to achieve its missions. The basic autonomy level is to maintain its stability following the desired path under embedded guidance, navigation, and control algorithm. The rapid advances in computing power, the efficiency of the DC motors, smaller microprocessors, the development of batteries and gyroscopic, Accelerometer and SLAM (Simultaneous Localization and Mapping) technology have all led to a proliferation of Rover designs. The rover is an Unmanned Ground Vehicle with an intense mixture of Electronics, Mechanical, and mainly the principle of robotics. The rover internal architecture of the rover consists of a control unit, a mechanical unit, and an application unit. The control unit stores programs that process the commands from the control station and thus handles the communication of the rover. The control unit is also responsible for fetching environmental parameters. The mechanical unit consists of motors for the movement of the rover and a very capable sensor package that includes an RC receiver. Cameras, LiDAR sensor, a gas sensor, an Altitude sensor, GPS, Gyroscope, Compass, Sonar, IR sensor. a thermal sensor, etc. In control engineering one may study Classical Control, Modern Control, Robust and Adaptive Control theories. In designing SLAM and Computer Vision Based UGV the suggested control theory applied is robust control theory which states that a feedback system will be robust one if and only if one may mention the range of variations that may occur during process and in our case this range is of x/y coordinates. According to variations which will occur this again and again.

1.2 Objective:

- Designing and implementing a RC Rover.
- 3d mapping and localization using SLAM technology.
- Image recognition system for ground serve using YOLO.
- To measure the temperature and humidity using DHT Temperature Sensor.
- To detect the flammable gas and smoke using the MQ-2 gas sensor.
- Able to monitor each sensor data in real time and live video feedback system.
- Able to 4K recording with Gimbal support.
- Custom GPS tag based autonomous flight system.
- Weapon and objects detection system.
- Implementation of RTH using GPS.

1.3 Motivation:

Due to this project we can learn about unmanned ground vehicle (UGV) and the implementation of A.I. for controlling the rover. This expand the scope of Computer Science & Engineering education to include the manual control and autonomous flight using python library. The rover has many applications that we are interested to develop like mapping, monitoring environmental status (temperature, humidity, CO2), video streaming, autonomous flight. Especially in disaster and dangerous areas. It also opens up the possibilities to broaden the understanding and application of control system, artificial intelligence and GPS navigation as it applies to the rover. UGV are used for surveillance and reconnaissance by military and law enforcement as well as search and rescue mission in a risky environment. The larger UGV also provide a long duration cost effective platform for reconnaissance as well as weapon. Our premise was that if smaller and cheaper UGVs become available to non-technical people, available it in commercial and industrial applications or even monitoring dangerous areas like radioactive area or war zone.

1.4 Scope

1.4.1 Search and Rescue:

Earthquake Risk in Bangladesh and some parts of the world are very high. Recently scientists have come to recognize that Bangladesh is positioned at the juncture of several active tectonic plate boundaries. Moreover, it sits atop the world's largest river delta at close to sea level, facing both the risk posed by a quake and secondary risks of tsunamis and flooding in the quakes aftermath. Rescuing survivors from a building collapsed due to earthquakes has proven to be one of the toughest challenges in today's world. But in the context of a rescue team, it is even more challenging to detect the specific regions of a large building and identify victims. The terrain is extremely flustered and the regions for exploration are often irregular in nature and very constrained. And these problems always make human rescue teams difficult to accomplish their tasks. UGVs can be used to traverse and 3d map mine tunnels. Combining Optical and sonar sensors, UGVs are developed to map 3D rock surfaces in open pit mines. And can detect workers in collapsed tunnels and can precisely calculate if the human rescuer can operate on that area or not. so in order to prevent, or at least minimize the loss of human lives UGVs are deployed on the scene. A low-cost search and rescue rover will help the human rescue team to plan their operation adequately.

1.4.2 Usefulness in the Context of Bangladesh:

We should be prepared for any approaching calamity because Bangladesh is a densely populated country with many natural risks. We can seek and rescue humans at any dangerous place faster and more efficiently if we have some rescue robots, and we can save more lives.

1.4.2.1 Bangladesh's Earthquake Risk:

Scientists have recently discovered that Bangladesh lies at the crossroads of many active tectonic plate boundaries. Furthermore, it is located at sea level atop the world's biggest river delta, putting it at danger from both an earthquake and the secondary risks of tsunamis and flooding that may occur as a result of the event. One of the most difficult difficulties in today's society is rescuing survivors from a structure that has collapsed due to earthquakes. However, detecting particular parts of a huge structure and identifying casualties is significantly more difficult in the context of a rescue squad. The landscape is exceedingly erratic, and the exploring areas are frequently uneven and limited. And these issues make it impossible for human rescue crews to complete their missions. A low-cost search and rescue robot will assist human rescuers in properly planning their mission.

1.4.2.2 Rana Plaza Incident:

On April 24, 2013, Rana Plaza, a nine-story commercial skyscraper in Savar, Dhaka, collapsed. The fall resulted in the deaths of about 1134 garment workers and the disappearance of hundreds more. The rescue attempts for this nine-story skyscraper took 20 days, but the rescue efforts for the World Trade Center took just 10 days since they had a specialist crew with robotics to assist and speed up the search process. If we had some rescue robots on hand during the Rana Plaza catastrophe, we might have accomplished the search mission in shorter time and so saved more lives.

1.4.3 Mobile mapping in dynamic environments:

UGVs with LiDAR are used to examine the surface of the earth, assess information about the ground surface, create a digital twin of an object or detail a range of geospatial information. LiDAR systems harness this technology, using LiDAR data to map three-dimensional models and digital elevation. Many of today's applications require accurate and fast 3D reconstructions of large-scale environments such as industrial plants, critical infrastructure (bridges, roads, dams, tunnels), public buildings, etc. These 3D reconstructions allow for further analysis of the scene, e.g. to detect wear or damages on the road surface or in tunnels. The 3D models can also be used to organize or monitor events in conference venues or in other event halls. Finally, in the domain of intelligent vehicles, 3D maps of the environment can facilitate autonomous driving. Unfortunately, the current process of 3D mapping is still an expensive and time-consuming process as it is often done using static laser scanning, hence needing a lot of different viewpoints and a lot of manual intervention and tuning.

1.4.4 Military and law Enforcement:

Unmanned ground vehicles (UGV) will play a key role in the Army's Objective Force structure. UGVs would be used for weapons platforms, logistics carriers, and reconnaissance, surveillance, and target acquisition among other things. UGV are used for surveillance and reconnaissance by military and law enforcement agencies (FBI, CIA, KGB). It use as well as search and rescue missions silently in urban and risky environments. Military heavy equipment for moving dirt under enemy fire, such as repairing craters in a runway or breaching a minefield or other barrier. Within a decade technological advances UGV could leave human operators out of the kill chain. UGVs Teleoperation capabilities, or the ability for an operator to manipulate and control remotely from a safe location through a tether or radio link, is the most mature control technology available and therefore is an area of emphasis for all Services in developing first generation rover programs. Teleoperation capabilities are important to the warfighter because they enable standoff operations and thereby reduce or remove operator risks in highly stressful and dangerous environments, such as minefields and in areas of potential explosive hazards.

1.4.5 Weapon and Object Detection:

The major problem of Rover of the warzone not being able to tell the difference between a child holding out an ice cream and someone holding a weapon. So with the help of AI and Deep Algorithms Rover can accurately detect humans and weapons of different types.

1.4.6 Research platform:

Unmanned ground vehicle is a land-based counterpart to remotely operate underwater vehicles and unmanned aerial vehicles. They are used in various missions where it may be inconvenient, dangerous, or impossible to have a human operator present. Rovers are a beneficial gadget for researchers to test and evaluate new ideas in different fields, AI implementations. There are various advantages of using rovers. They are relatively cheap, available in a different sizes and their straightforward mechanical design means can be built and maintained by amateurs.

1.4.7 Civilian and commercial applications:

Multiple civilian applications of UGVs are being implemented to automatic processes in manufacturing and production environments. They have also been developed as autonomous tour guides for the Carnegie Museum of Natural History and the Swiss National Exhibition Expo.

1.4.8 Emergency response:

UGVs are used in many emergency situations including Urban search and rescue, fire-fighting, and nuclear response. Following the 2011 Fukushima Daiichi Nuclear Power Plant accident, UGVs were used in Japan for mapping and structural assessment in areas with too much radiation to warrant a human presence.

1.4.9 Fire Hazard:

Fire, lightning, hailstorm, storm, nor wester, building collapse, bridge collapse, boat capsize, and wild animal assault were among the nine severe disasters that struck Bangladesh in March 2019. Fire was the most common danger this month, with 24 incidences reported in Dhaka, Chattogram, Madaripur, Chandpur, Rangpur, Lakshmipur, Cumilla, Jhalokathi, Gopalganj, Gazipur, and Noakhali, respectively. If the ugvs were implemented on those rescue mission many lives could’ve been saved.

1.4.10 Toxic Gas Release:

Toxic gases (or noxious gases) are gases that are harmful to living things. They can easily build up in confined working spaces when the production process uses noxious gases. It may also result in the biological chemical breakdown of a substance that is being stored in a tank. Certain factory-related activities, such as welding, can also result in the build-up of toxic gases in a confined space. UGVs can detect hazardous gases with gas and thermal sensors. So with the help of those sensors the ugv can determine the safety of the buildings or in those specific region where the ugv is deployed.

1.4.11 Journalism:

Major media outlets and many military organizations have started using Rovers for reporting and verifying news on events that include natural disasters, live telecasting of warzones.

1.4.12 Humanitarian operations:

Rovers are being used for humanitarian applications from disaster relief to animal conservation. During flood and earthquake, it is use to delivery necessary payload and medicine for fast response.

Chapter 2

Literature Review

The earliest robots as we know them were created in the early 1950s by George C. Devol, an inventor from Louisville, Kentucky. He invented and patented a reprogrammable manipulator called “Unimate,” from “Universal Automation.” For the next decade, he attempted to sell his product in the industry, but did not succeed.One may find many research papers based on computer vision and Lane detection whereas only few of them deal with non-marked paths. This technique is mostly used for autonomous vehicles. Since one is pursuing for designing and implementing the autonomous vehicle which means this vehicle will be equipped with several different sensors which will assist the system to deal with grey scale and segmented images. After going through various road recognition algorithms and approaches one may find the majority of them are based on edge detection whereas the approaches based on texture and Gabor wavelets are also available. Various authors used stereo vision along with additional sensors like LIDAR for detecting curbs. Discussing the technical term of Path Planning, the task of the searching is the basic actions based on a series of sequence that will lead an autonomous robot to reach its final state whereas discussing in the context of robotic rovers, many papers will suggest to represent the environment in terms of map and a set of fan ins (sensor inputs). Many research publications are suggesting various basic components i.e. GPS, velocity meter, electronic thermometer and manometer etc. whereas sonar and camera has been suggested for recognition of any hurdle. There is typically one on one mapping of images and image arrays so that location or more technically image space occupancy grid locations can be identified. However, the path or image space planning is basically assumed as a function in between autonomous vehicles and image coordinate which can be easily achieved directly if distance value is known. In addition to this, the common assumption in designing this thing is to assume the robot is situated on a flat ground. During a detailed research review one may also come across sequenced route representation which uses the relationship between a target image and camera’s current captured image to calculate the correlation. Utilizing this technique one may correlate many things i.e. lane markings, prior vehicle tires or sign boards on tracks. Studying about the behavior of one robot it may follow path from Cartesian space along with incorporation of way points conveyed to it from image space paths which are not yet achieved and can be sufficiently a good method. Whereas the path planner is also categorized into 2 subsystems i.e. parallel operation and a common feature extraction based system in order to make design faultless. Path planning is an important primitive for autonomous mobile robots. Furthermore, lets to the robots find the shortest or otherwise optimal path between two points. Otherwise optimal paths could be paths that minimize the amount of turning, the amount of braking or whatever a specific application requires. During the literature review many of the papers refer two proposed methods for hierarchical path planning i.e. Image Space Verification of Global Planning IVG and Fast Local Cylindrical Planning FLCP, both are based on the Cartesian coordinate system.

UGVS are an active and growing research area. This has led to developments in path planning algorithms. Many algorithms have been proposed to address the issues of UGV path planning. Based on its solution method, path planning algorithms can be classified as either exact (deterministic) methods or approximate methods. Exact methods are enumerative and follow systematic steps to find the solution. The majority of classical/conventional algorithms fall in this category. Approximate methods can be divided into two categories, approximation algorithms, and heuristic algorithms. Approximation algorithms produce "provable solution quality and provable run-time bounds." Heuristic algorithms provide good solutions for large-instance problems in a reasonable amount of time. Therefore, accepted performance is obtained within an acceptable cost. Heuristic algorithms are further classified into meta heuristics and problem-specific heuristics. Meta-heuristics are applicable to a wide range of optimization problems without any adaptation to the specific problem, such as ant colony optimization (ACO) and simulated annealing (SA). Problem-specific heuristics are designed to find a solution to a specific problem.

Nature-inspired algorithms, which are considered meta-heuristic, belong to a class of algorithms that take advantage of the perfection of natural systems. Thus, nature-inspired algorithms are capable of efficiently solving very complex problems. They can be categorized into physics-inspired, biology-inspired, and chemistry-inspired algorithms.

The shortest path is one of the most known problems and brings a lot of interest in Early history of shortest paths algorithms which are presented by Shimbel in 1955. Information networks. Ford (1956), RAND, economics of transportation, Leyzorek, Gray, Johnson, Ladew, Meaker, Petry, Seitz (1957), Combat Development Dept. of the Army Electronic Proving Ground. Dantzig (1958), Simplex method for linear programming, Bellman (1958). Dynamic programming. Moore (1959), Routing long-distance telephone calls for Bell Labs, Dijkstra (1959), Simpler and faster version of Ford’s algorithm. The Shortest path problem is to find a distance between two vertices Shortest path problem is about finding a path between two vertices in a graph such that the total sum of the edge weights is minimized. This problem could be solved easily when using all edge weights, but here the weights can take any value. The use of algorithms which find the shortest paths are important and not only in robotics, but also in network routing, video games and gene sequencing. The shortest path problems are by far the most fundamental and also most commonly encountered problems in the study of transportation and communication networks. In few of the papers proposed one may find a technique that evaluates the fixed set of possible trajectories and assumes the presence of any scene i.e. pedestrian footprints etc. In contrast to IVG and FLCP this technique produces less efficient results. The discussion related to shortest path problems and computation can be solved using either Kruskal's or Dijkstra's algorithms which are similar to two-point distance formula but are critical to understand and manipulate. This technique can mostly be found in the papers suggested for designing the vehicle robot for military purposes. One may also come across the term of disparity during studying different research pieces; it simply refers to the 1D displacement of corresponding the stereo image pair which is inversely related to the distance of an observed hurdle. In some of the current publications, the major parameters are computed using Radon transformation using central slice theorem. One of the technical terms is Visual odometer which refers to the process of estimating the 3D motion of a vehicle. The algorithm defined for the mentioned term is designed basically to identify the features of interest FOI in each camera frame, estimate depth of each feature (typically using stereo), match features across time frames, and then estimate the rigid body transformation that best aligns the features over time. Since then, a great deal of progress has been made in all aspects of visual odometer as well as in features of unmanned vehicles that are controlled by GPS coordination. In such features one may find Harris corners along with FAST Features being included and this technique is comparatively quick to compute the resilient changes.

Chapter 3

History of Autonomous UGV.

3.1 History:

French engineers in 1915 developed the “Torpille Terrestre” (Land Torpedo), a breeching UGV loaded with explosives. A working remote controlled car was reported in the October 1921 issue of RCA's World Wide Wireless magazine. The car was unmanned and controlled wirelessly via radio, it was thought the technology could someday be adapted to tanks. In the 1930s, the USSR developed Teletanks, a machine gun-armed tank remotely controlled by radio from another tank. These were used in the Winter War (1939-1940) against Finland the Teletanks could be mounted with machine guns, flamethrowers, and bombs. Closely thereafter, the German army unveiled the Goliath, a French-designed remote-control mini-tank that could be driven towards targets and remotely detonated. Despite their humble capabilities and their tendency to break, 7,564 Goliaths were produced. The Second World War saw the development of a number of similar single-use weapons platforms. Though it is tempting to define these as UGVs, the fact that they served exclusively single use bombs, rather than weapons delivery systems, means that they were more like precision-guided munitions than unmanned ground vehicles.

At the start of the Eastern Front after Germany invaded the USSR in 1941. During World War II, the British developed a radio control version of their Matilda II infantry tank in 1941. Known as "Black Prince", it would have been used for drawing the fire of concealed anti-tank guns, or for demolition missions. Due to the costs of converting the transmission system of the tank to Wilson type gearboxes, an order for 60 tanks was cancelled. From 1942, the Germans used the Goliath tracked mine for remote demolition work. The Goliath was a small tracked vehicle carrying 60 kg of explosive charge directed through a control cable. Their inspiration was a miniature French tracked vehicle found after France was defeated in 1940. The combination of cost, low speed, reliance on a cable for control, and poor protection against weapons meant it was not considered a success. Many types of UGVs are developed with heavy armed weapons and payloads packing with many features.

3.2 Revolution:

The development of autonomous robots began as an interesting application domain for Artificial Intelligence researchers in the late 1960s.The first major mobile robot development effort was SHAKY 2 developed in the late 1960s to serve as a testbed for DARPA-founded artificial intelligence. SHAKEY was a wheeled platform equipped with steerable TV camera, ultrasonic range finder, and touch sensors, connected via an RF link to its mainframe computer that performed navigation and exploration tasks. The SHAKEY system could accept English sentence commands from the terminal operator, directing the robot to push large wooden blocks around in its lab environment "world". The action routines took care of simple moving, turning, and route planning. The programs could make and execute plans to achieve goals given to it by a user.

The SHAKEY program reemerged in the early 1980s as the DARPA Autonomous Land Vehicle (ALV). Under DARPA's Strategic Computing (SC) Program. The Autonomous Land Vehicle was built on a Standard Manufacturing eight-wheel hydrostatically-driven all-terrain vehicle capable of speeds of up to 45 mph on the highway and up to 18 mph on rough terrain. The ALV could carry six full racks of electronic equipment in dust-free air conditioned comfort, providing power from its 12-kW diesel power unit. The initial sensor suite consisted of a colour video camera and a laser scanner from the Environmental Research Institute of Michigan. The ALV Program's focus was moved in early 1988 away from integrated demonstrations of military applications and toward the support of specific scientific experiments for off-road navigation.

The Reconnaissance, Surveillance and Target Acquisition (RSTA) application has long drawn the attention of UGV developers, since a UGV solution for RSTA would provide a battlefield commander with a direct sensing capability on the battlefield and even behind enemy lines, without endangering human personnel. Two RSTA-oriented UGV projects were undertaken at the Naval Ocean Systems Centre (NOSC) in the early 1980s. The Ground Surveillance Robot (GSR) at NOSC San Diego, and the Advanced Teleoperator Technology (ATT) TeleOperated Dune Buggy at NOSC Hawaii. The Ground Surveillance Robot project explored the development of a modular, flexible distributed architecture for the integration and control of complex robotic systems, using a fully actuated 7-ton M-114 armoured personnel carrier as the testbed host vehicle. With an array of fixed and steerable ultrasonic sensors and a distributed blackboard architecture implemented on multiple PCs, the vehicle successfully demonstrated autonomous following of both a lead vehicle and a walking human in 1986 before funding limitations terminated its development. The Advanced Teleoperator Technology TeleOperated Dune Buggy, on the other hand, concentrated exclusively on teleoperator control methodology and on "advanced, spatially-correspondent multi-sensory human/machine interfaces." With a Chenowth dune buggy as a testbed vehicle, the Advanced Teleoperator Technology project successfully demonstrated the feasibility of utilising a remotely operated ground vehicle to transit complex natural terrain and of remotely operating vehicle-mounted weapons systems. In addition, the Advanced Teleoperator Technology effort demonstrated the efficacy of stereo head-coupled visual display systems, binaural audio feedback, and isomorphic vehicle controls for high-speed remote vehicle operations.

3.3 Unmanned Ground Vehicles:

The Unmanned Ground Vehicles Joint Program Office (UGV JPO) was organised as the centrum for the development and fielding of DoD unmanned ground vehicles systems in 1990. The word "systems" was later added to the program office name, so the organisation is the Unmanned Ground Vehicles/ Systems Joint Program Office (UGV/S JPO.) a demonstration of a near-term teleoperated UGV capabilities and technologies led by the Army Research Laboratory in the spring of 1992. DEMO II demonstrated multiple vehicles operating cooperatively under supervised autonomy.

The second milestone DEMO B was on 28–30 June 1994. The Surrogate Semi- autonomous Vehicle, in a supervised-autonomous mode, successfully transitioned from road-following on a dirt road to road-following on a paved road and then to a cross-country route without stopping. In the cross-country mode, the SSV transmitted target data back to the operator The last milestone DEMO C was the final DEMO II demonstration during 25–28 July 1995. This was the first time that two Surrogate Semi-autonomous Vehicles (SSV) worked together in a scout mission. The two SSVs moved autonomously over a cross-country route negotiating obstacles at a speed of 5mph. Once the SSV's arrived at their designated location, the vehicles began a search of the area providing information back to the multi-vehicle operator control vehicle In August 1997 four Panthers and mini flails continued to support forces in Bosnia as part of Operation Joint Guard. The Panthers were used to proof an area.

The remote ordnance neutralisation vehicles play an important role in the family of unmanned ground vehicles. The Remote Ordnance Neutralisation (RON)10 program received and tested its first system in 1999 The latter part of the 20th century saw the advent of the 'digital revolution', which resulted in dramatic advances in computing processing power, sensor technology and satellite telecommunications. These technical developments permitted a commensurate evolution in UGV independence and autonomy and by the turn of the century, technology was sufficiently advanced to generate real interest in deploying UVs outside of covert military operations. However, it was perhaps the terrorist attacks in September 2001 in the United States that served as the most important catalyst for the adoption of UGVs as a key counterinsurgency tool. Of particular note is the ability of UGVs to provide global, persistent surveillance; reduce the sensor-to-shoot cycle; and undertake dull dirty and dangerous roles. These factors are discussed in greater detail below:

The DARPA Urban Challenge held in November 2007 was their latest competition. The rules were similar but this time the teams were required to build an autonomous vehicle capable of driving in traffic, performing complex manoeuvres such as merging, passing, parking and negotiating intersections. This event was truly groundbreaking as it was the first time that autonomous vehicles had interacted with both manned and unmanned vehicle traffic in an urban environment. After a qualification event it was announced that eleven teams were selected to compete. The event finished with six teams crossing the line, the winner being Tartan Racing, from Carnegie Mellon, Pittsburgh who demonstrated the best performance. One of the most common and impactful applications of UGVs is their use as planetary rovers. Starting with the Soviet moon exploration rover Lunokhod 1, planetary rovers have been in use for well over 30 years. The most recent rover of interest is Curiosity, which was launched in 2011. For a full review of both Lunar and Martian rovers.

There has been many types of UGVs in recent years .in recent days Boston dynamics is one of the well-known developers and researchers of UGVs. Their most recent work in UGVs are On June 23, 2016, Boston Dynamics four-legged canine-inspired spot.

There was some demonstration of programs and some interesting models of UGV development from USA. It was not complete. It was only a flash from about one hundred programs. In my opinion the research and development of unmanned ground vehicles is very interesting and very important segment of military technical modernisation. We have to theoretically follow the significant foreign efforts, and from time to time review their products. At the end we want to cite one sentence from STAR 21 report, Strategic Technologies of the Twenty-First Century: “The core weapon of the 20 th century has been the tank. The core weapon of the 21st century may well be the unmanned systems”.

Chapter 4
Neural Network and CNN

4.1 Neural Networks:

Neural networks are a set of algorithms, modeled loosely after the human brain, that are designed to recognize patterns. They interpret sensory data through a kind of machine perception, labeling or clustering raw input. The patterns they recognize are numerical, contained in vectors, into which all real-world data, be it images, sound, text or time series, must be translated. Neural networks help us cluster and classify. They help to group unlabeled data according to similarities among the example inputs, and they classify data when they have a labeled dataset to train on. (Neural networks can also extract features that are fed to other algorithms for clustering and classification; so you can think of deep neural networks as components of larger machine-learning applications involving algorithms for reinforcement learning, classification and regression.

A single neuron can perform only a simple task—it is either on or off. Complex functions can be designed and performed using a network of interconnecting neurons or perceptions. The structure of a network can be complicated, and one of the most widely used is to arrange them in a layered structure, with an input layer, an output layer, and one or more hidden layers. The connection strength between two neurons is represented by its corresponding weight. Some artificial neural networks (ANNs) can perform complex tasks and can simulate complex mathematical models, even if there is no explicit functional form mathematically. Neural networks have been developed over the last few decades and applied in almost all areas of science and engineering. The construction of a neural network involves the estimation of the suitable weights of a network system with some training/known data sets.

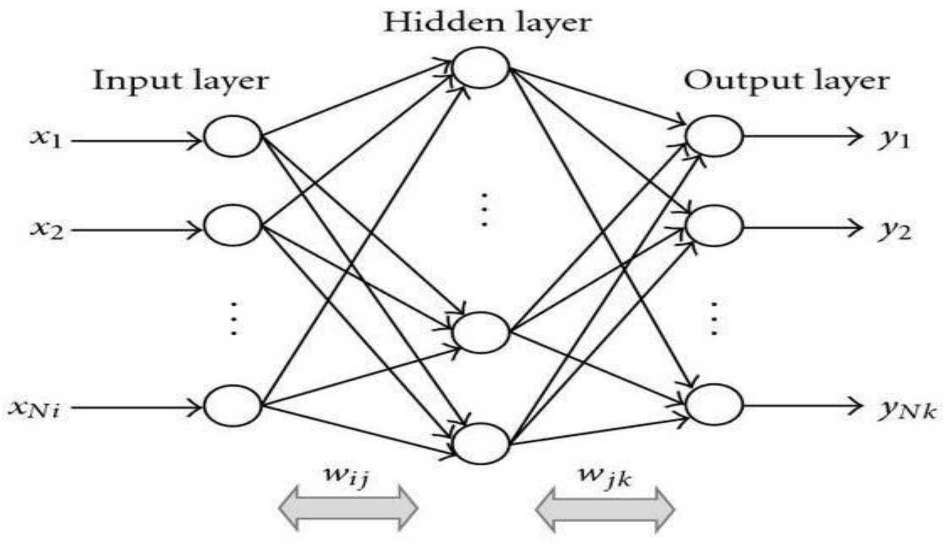


Figure 1: Schematic representation of a three-layer neural network.

The task of the training is to find the suitable weights w_{ij} such that the neural networks not only can best-fit the known data but also can predict outputs for new inputs. A good artificial neural network should be able to minimize both errors simultaneously—the fitting/learning errors and the prediction errors.

4.2 CNN:

CNNs are powerful image processing, artificial intelligence (AI) that use deep learning to perform both generative and descriptive tasks, often using machine vison that includes image and video recognition, along with recommender systems and natural language processing (NLP).CNN's were first developed and used around the 1980s. The most that a CNN could do at that time was recognize handwritten digits. It was mostly used in the postal sectors to read zip codes, pin codes, etc.

A neural network is a system of hardware and/or software patterned after the operation of neurons in the human brain. Traditional neural networks are not ideal for image processing and must be fed images in reduced-resolution pieces. CNN have their "neurons" arranged more like those of the frontal lobe, the area responsible for processing visual stimuli in humans and other animals. The layers of neurons are arranged in such a way as to cover the entire visual field avoiding the piecemeal image processing problem of traditional neural networks. A CNN uses a system much like a multilayer perceptron that has been designed for reduced processing requirements. The layers of a CNN consist of an input layer, an output layer and a hidden layer that includes multiple convolutional layers, pooling layers, fully connected layers and normalization layers. The removal of limitations and increase in efficiency for image processing results in a system that is far more effective, simpler to trains limited for image processing and natural language processing.

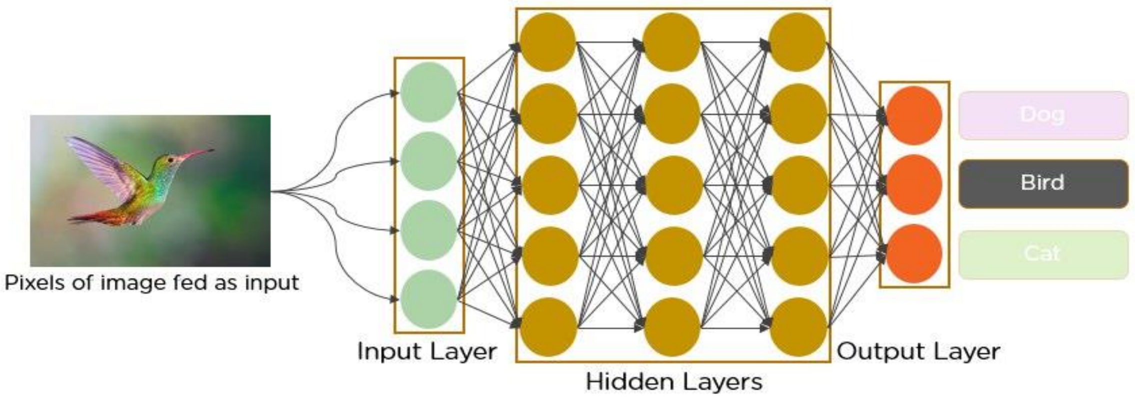


Figure 2: Layers of CNN

The important thing to remember about any deep learning model is that it requires a large amount of data to train and also requires a lot of computing resources. This was a major drawback for CNNs at that period and hence CNNs were only limited to the postal sectors and it failed to enter the world of machine learning.

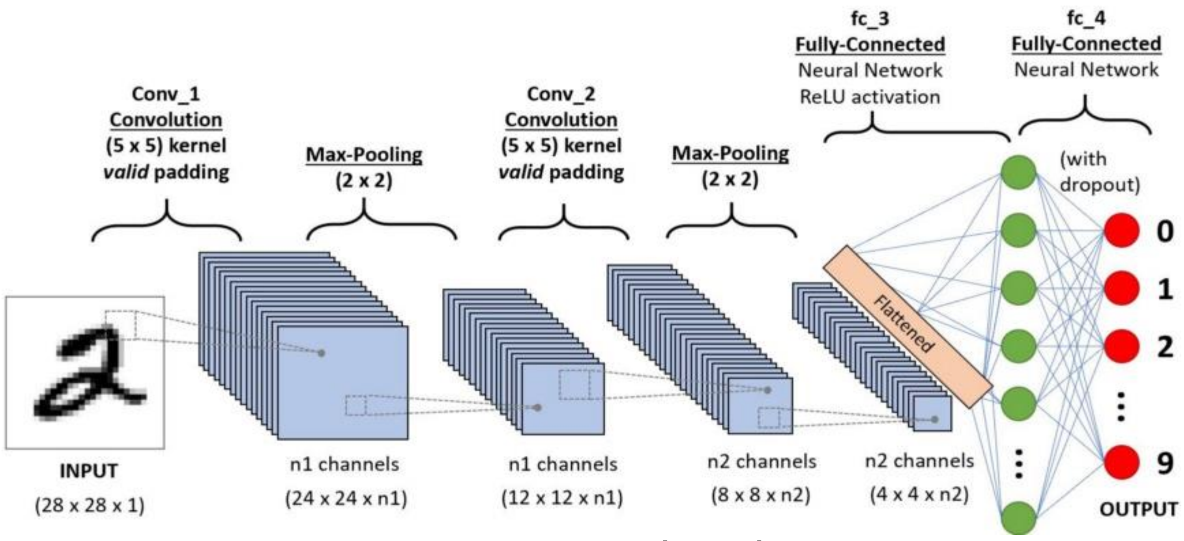


Figure 3: Deep Neural Network.

In deep learning, a convolutional neural network (CNN/ConvNet) is a class of deep neural networks, most commonly applied to analyze visual imagery. Now when we think of a neural network we think about matrix multiplications but that is not the case with ConvNet. It uses a special technique called Convolution. Now in mathematics convolution is a mathematical operation on two functions that produces a third function that expresses how the shape of one is modified by the other.

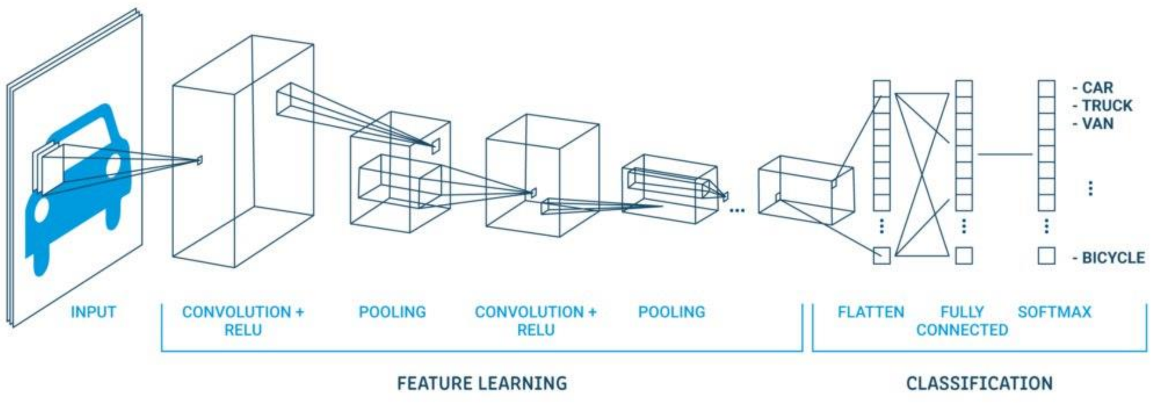


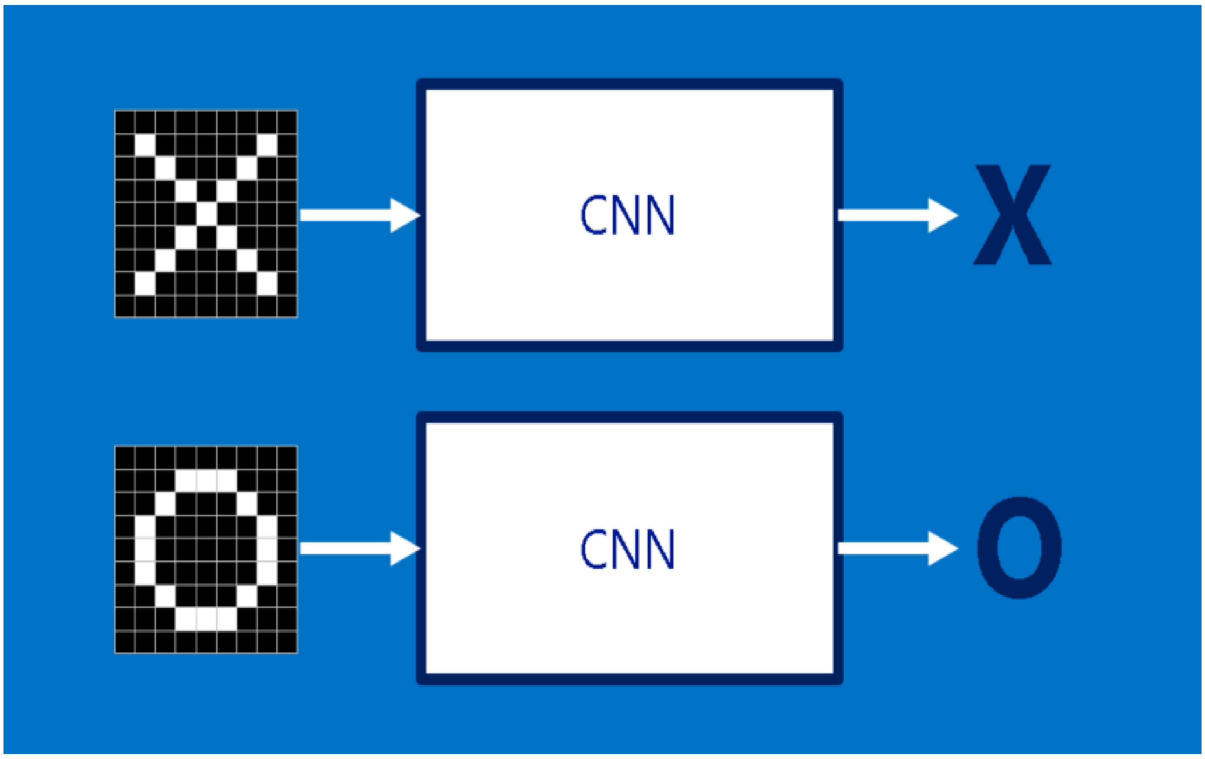
Figure 4: Overall process of CNN.

But we don’t really need to go behind the mathematics part to understand what a CNN is or how it works. Bottom line is that the role of the ConvNet is to reduce the images into a form that is easier to process, without losing features that are critical for getting a good prediction.

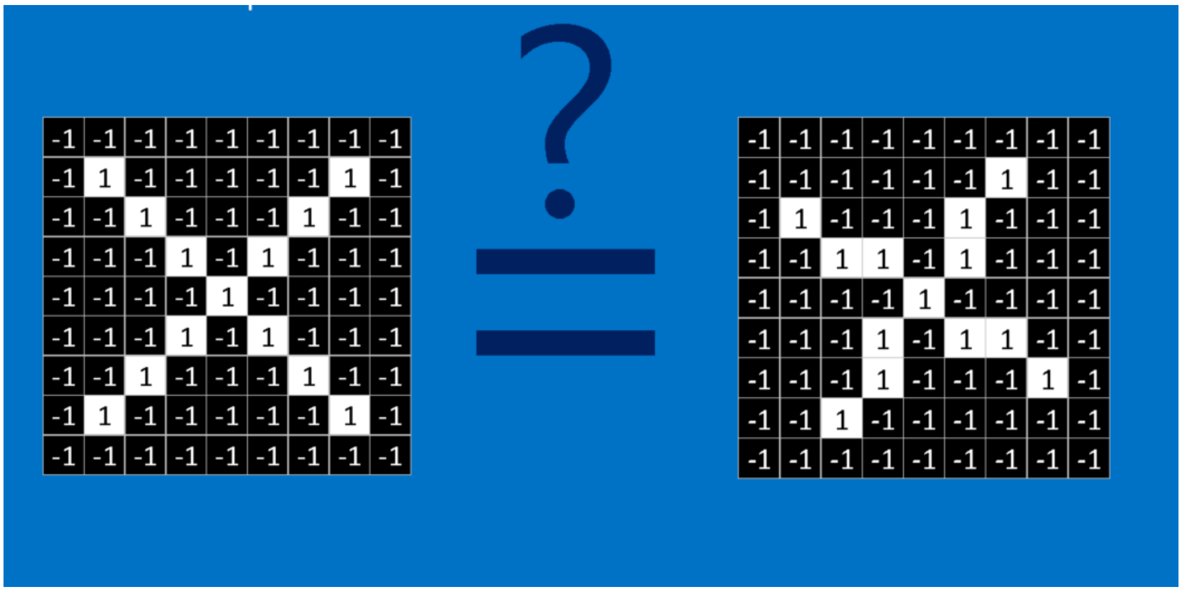
4.3 Working principle of CNN:

Generally, A Convolutional neural network has three layers. And we understand each layer one by one with the help of an example of the classifier. With it can classify an image of an X and O. So, with the case, we will understand all four layers.

X's and O's:

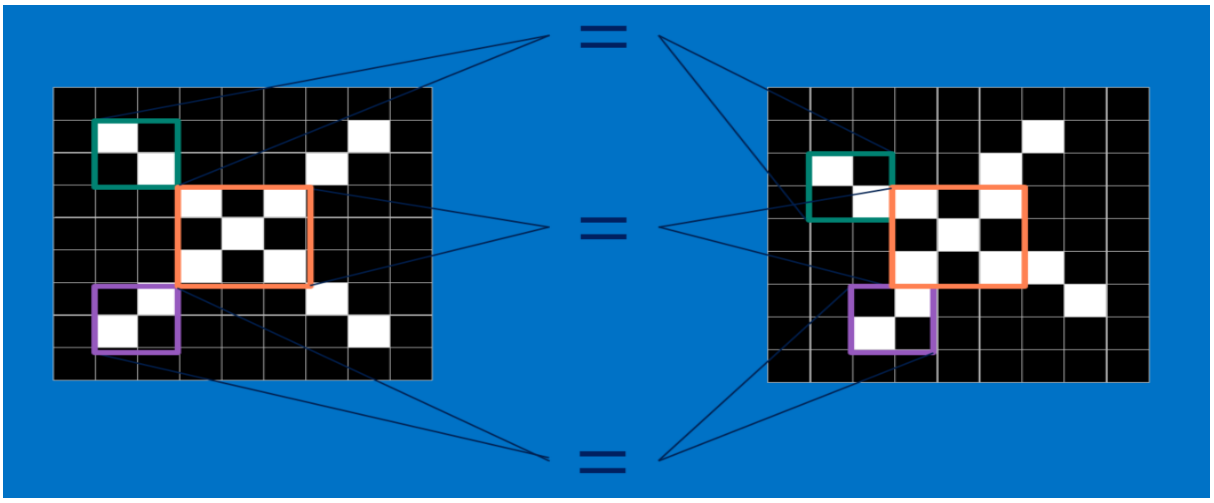


To help guide our walk through a Convolutional Neural Network, we’ll stick with a very simplified example: determining whether an image is of an X or an O. This example is just rich enough to illustrate the principles behind CNNs, but still simple enough to avoid getting bogged down in non-essential details. Our CNN has one job. Each time we hand it a picture, it has to decide whether it has an X or an O. It assumes there is always one or the other.

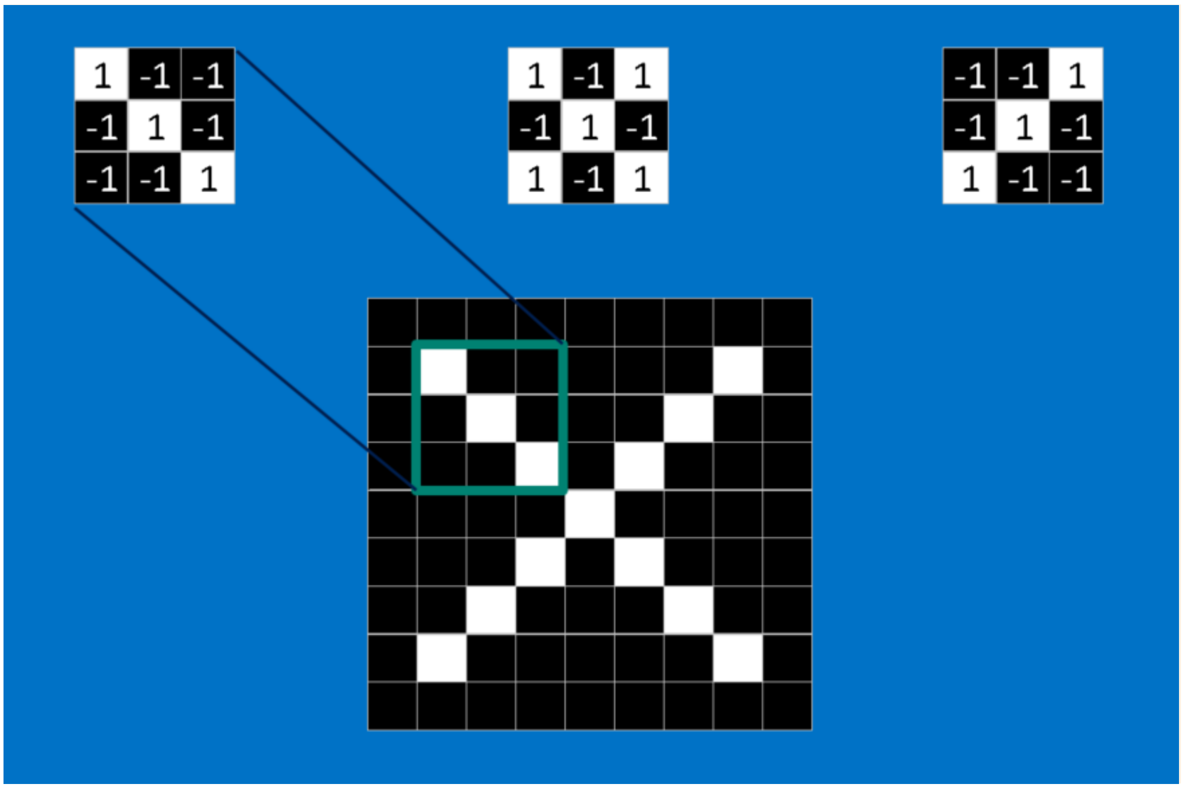


A naïve approach to solving this problem is to save an image of an X and an O and compare every new image to our exemplars to see which is the better match. What makes this task tricky is that computers are extremely literal. To a computer, an image looks like a two-dimensional array of pixels (think giant checkerboard) with a number in each position. In our example a pixel value of 1 is white, and -1 is black. When comparing two images, if any pixel values don't match, then the images don't match, at least to the computer. Ideally, we would like to be able to see X's and O's even if they're shifted, shrunken, rotated or deformed. This is where CNNs come in.

Features:



CNNs compare images piece by piece. The pieces that it looks for are called features. By finding rough feature matches in roughly the same positions in two images, CNNs get a lot better at seeing similarity than whole-image matching schemes.



Each feature is like a mini-image—a small two-dimensional array of values. Features match common aspects of the images. In the case of X images, features consisting of diagonal lines and a crossing capture all the important characteristics of most X's. These features will probably match up to the arms and center of any image of an X.

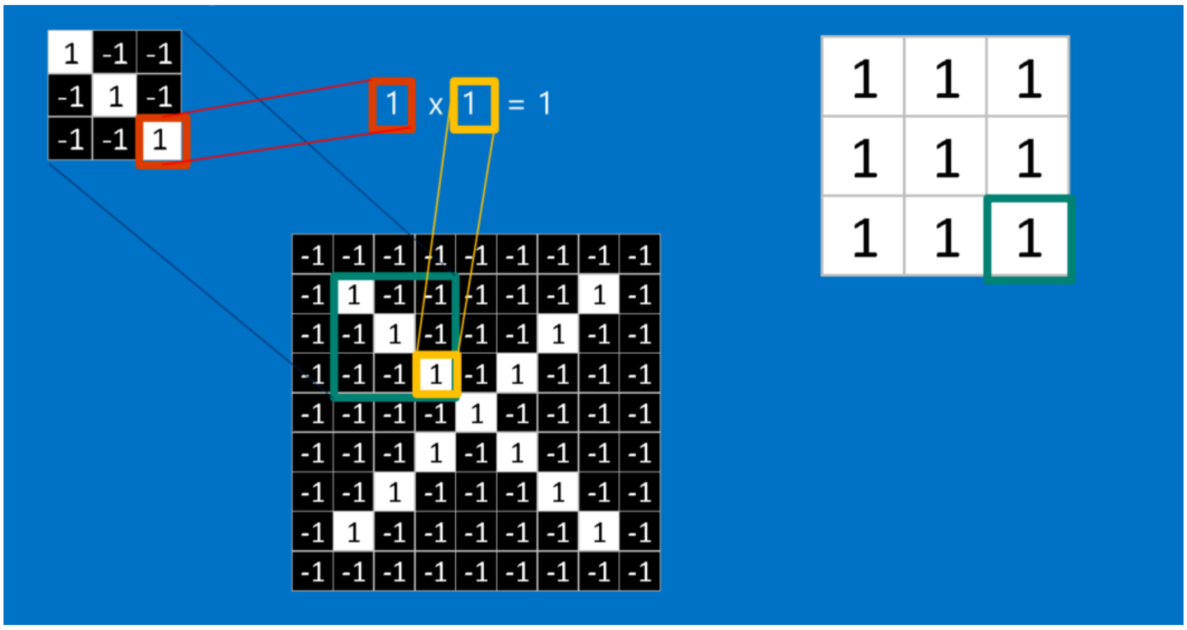
Convolutional Neural Networks have the following layers:

- Convolutional
- ReLU Layer
- Pooling
- Fully Connected Layer

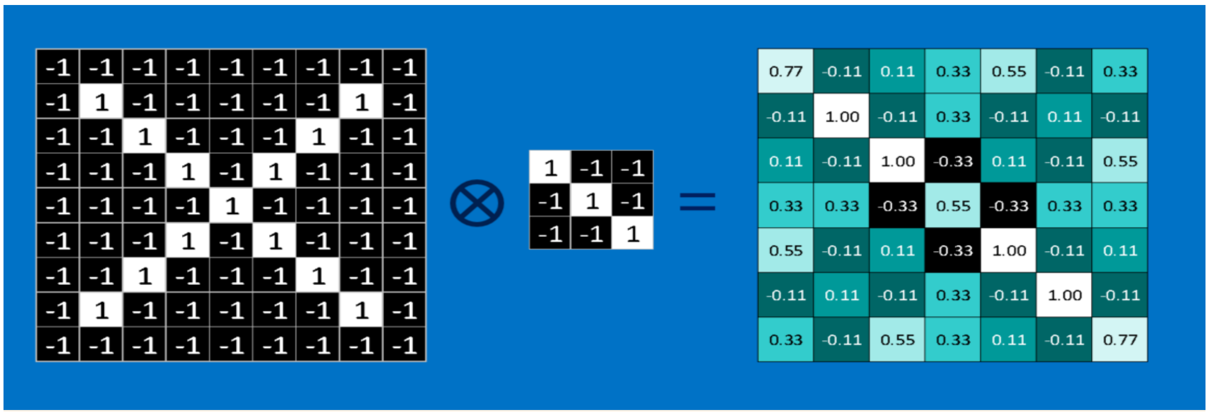
Before we go to the working of CNN's let's cover the basics such as what is an image and how is it represented. An RGB image is nothing but a matrix of pixel values having three planes whereas a grayscale image is the same but it has a single plane. Take a look at this image to understand more.

Convolution:

When presented with a new image, the CNN doesn't know exactly where these features will match so it tries them everywhere, in every possible position.

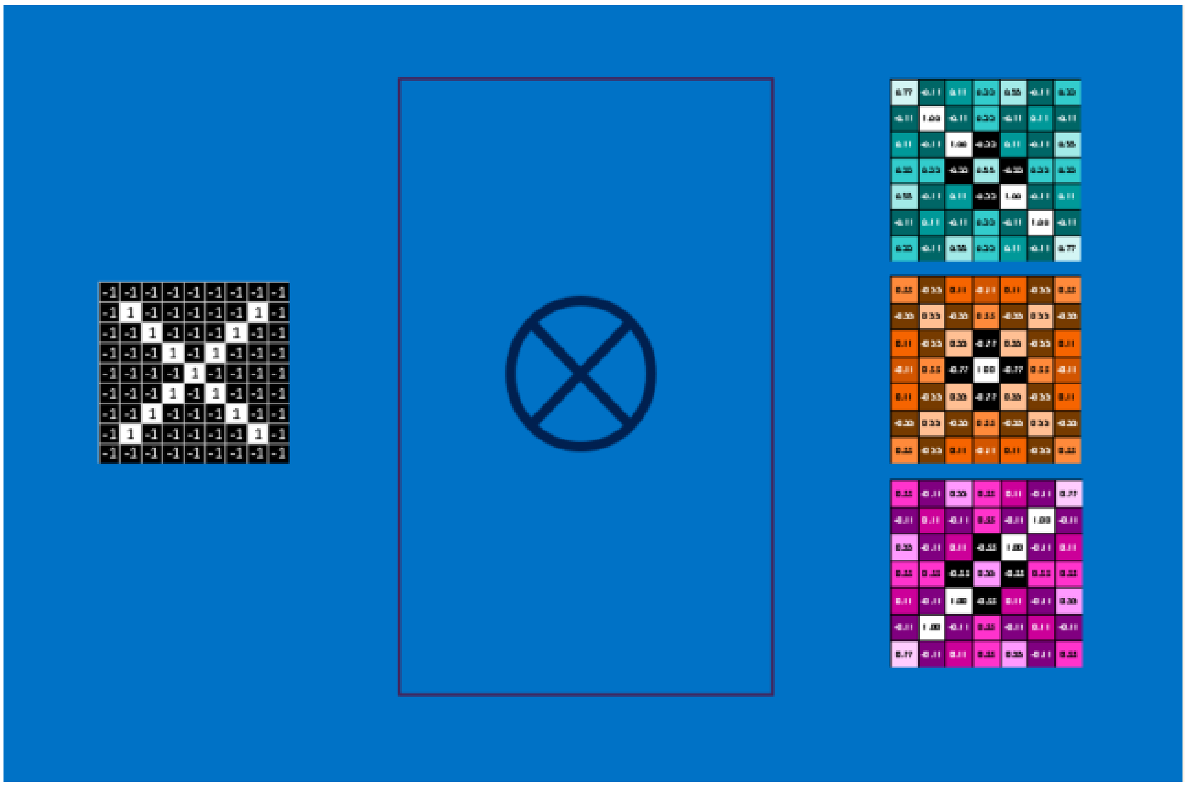


In calculating the match to a feature across the whole image, we make it a filter. The math we use to do this is called convolution, from which Convolutional Neural Networks take their name. The math behind convolution is nothing that would make a sixth-grader uncomfortable. To calculate the match of a feature to a patch of the image, simply multiply each pixel in the feature by the value of the corresponding pixel in the image. Then add up the answers and divide by the total number of pixels in the feature. If both pixels are white (a value of 1) then 1 * 1 = 1. If both are black, then (-1) * (-1) = 1. Either way, every matching pixel results in a 1. Similarly, any mismatch is a -1. If all the pixels in a feature match, then adding them up and dividing by the total number of pixels gives a 1. Similarly, if none of the pixels in feature match the image patch, then the answer is a -1.



$$\begin{bmatrix} -1&-1&-1&-1&-1&-1&-1&-1&-1\\ -1&1&-1&-1&-1&-1&-1&1&-1\\ -1&-1&1&-1&-1&-1&1&-1&-1\\ -1&-1&-1&1&-1&1&-1&-1&-1\\ -1&-1&-1&-1&1&-1&-1&-1&-1\\ -1&-1&-1&1&-1&1&-1&-1&-1\\ -1&-1&1&-1&-1&-1&1&-1&-1\\ -1&1&-1&-1&-1&-1&-1&1&-1\\ -1&-1&-1&-1&-1&-1&-1&-1&-1 \end{bmatrix} \otimes \begin{bmatrix} 1&-1&-1\\ -1&1&-1\\ -1&-1&1 \end{bmatrix} = \begin{bmatrix} 0.77&-0.11&0.11&0.33&0.55&-0.11&0.33\\ -0.11&1.00&-0.11&0.33&-0.11&0.11&-0.11\\ 0.11&-0.11&1.00&-0.33&0.11&-0.11&0.55\\ 0.33&0.33&-0.33&0.55&-0.33&0.33&0.33\\ 0.55&-0.11&0.11&-0.33&1.00&-0.11&0.11\\ -0.11&0.11&-0.11&0.33&-0.11&1.00&-0.11\\ 0.33&-0.11&0.55&0.33&0.11&-0.11&0.77 \end{bmatrix}$$

To complete our convolution, we repeat this process, lining up the feature with every possible image patch. We can take the answer from each convolution and make a new two dimensional array from it, based on where in the image each patch is located. This map of matches is also a filtered version of our original image. It’s a map of where in the image the feature is found. Values close to 1 show strong matches, values close to -1 show strong matches for the photographic negative of our feature, and values near zero show no match of any sort.



The next step is to repeat the convolution process in its entirety for each of the other features. The result is a set of filtered images, one for each of our filters. It’s convenient to think of this whole collection of convolution operations as a single processing step. In CNNs this is referred to as a convolution layer, hinting at the fact that it will soon have other layers added to it.

It’s easy to see how CNNs get their reputation as computation hogs. Although we can sketch our CNN on the back of a napkin, the number of additions, multiplications and divisions can add up fast. In math speak, they scale linearly with the number of pixels in the image, with the number of pixels in each feature and with the number of features. With so many factors, it’s easy to make this problem many millions of times larger without breaking a sweat. Small wonder that microchip manufacturers are now making specialized chips in an effort to keep up with the demands of CNNs.

Pooling:

Another power tool that CNNs use is called pooling. Pooling is a way to take large images and shrink them down while preserving the most important information in them.

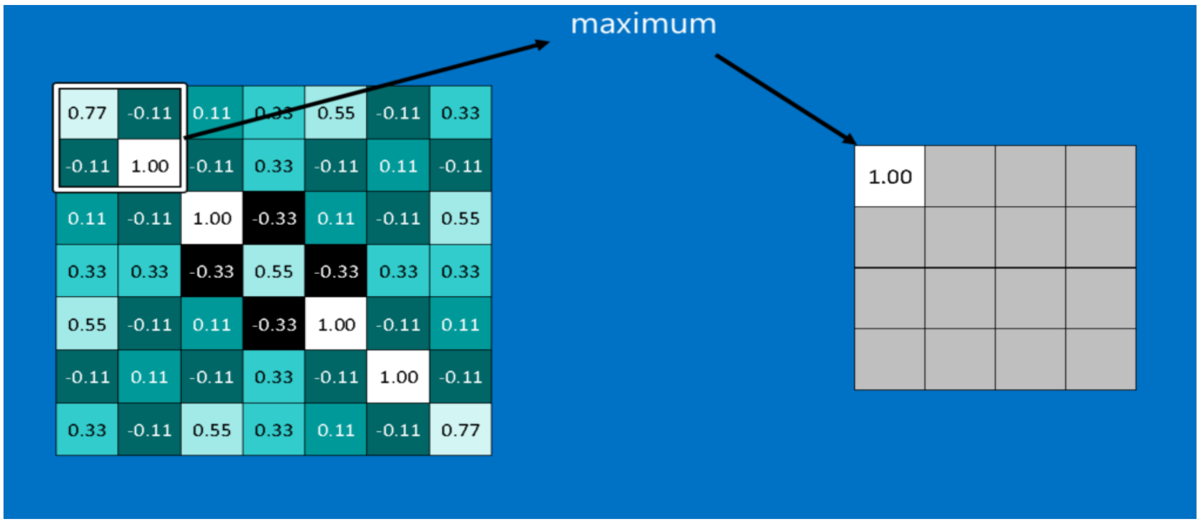
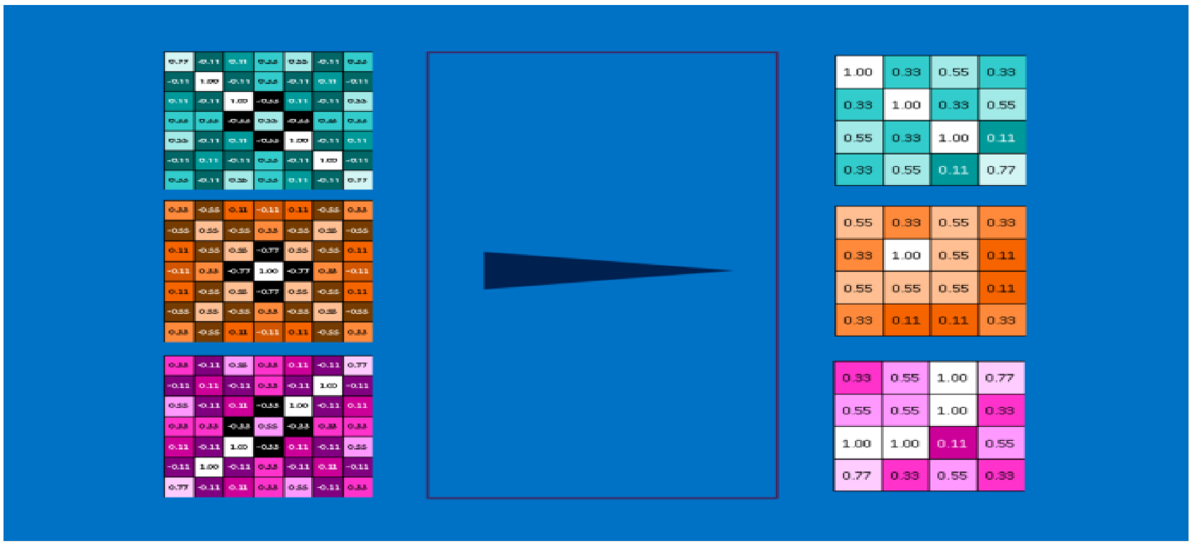


Figure 5: Pooling

The math behind pooling is second-grade level at most. It consists of stepping a small window across an image and taking the maximum value from the window at each step. In practice, a window 2 or 3 pixels on a side and steps of 2 pixels work well. After pooling, an image has about a quarter as many pixels as it started with. Because it keeps the maximum value from each window, it preserves the best fits of each feature within the window. This means that it doesn't care so much exactly where the feature fit as long as it fit somewhere within the window. The result of this is that CNNs can find whether a feature is in an image without worrying about where it is. This helps solve the problem of computers being hyper-literal.

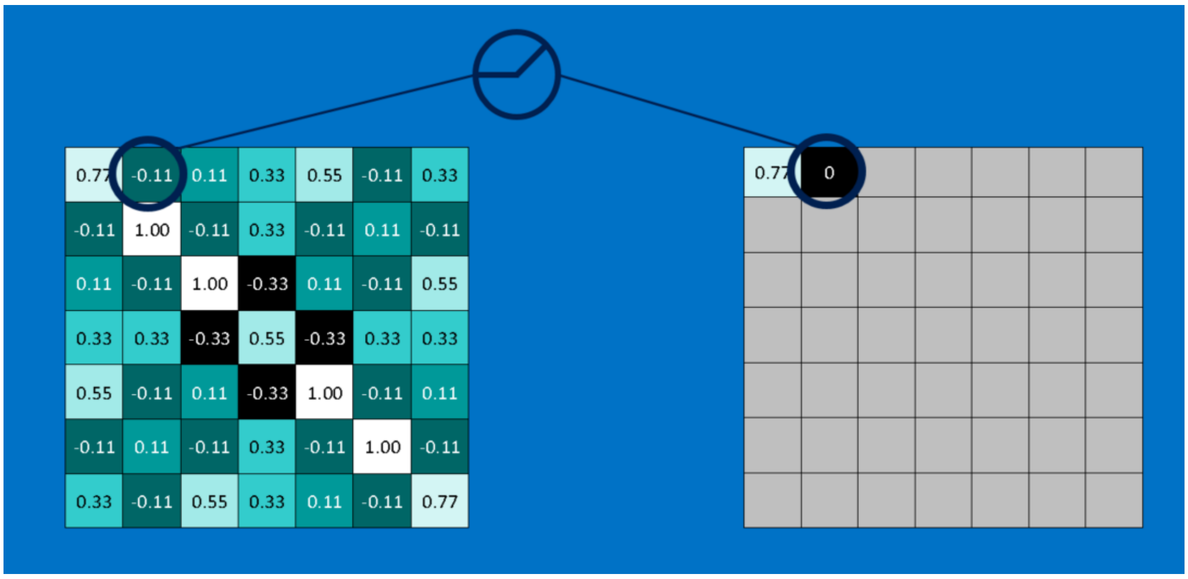


A pooling layer is just the operation of performing pooling on an image or a collection of images. The output will have the same number of images, but they will each have fewer pixels.

This is also helpful in managing the computational load. Taking an 8 megapixel image down to a 2 megapixel image makes life a lot easier for everything downstream.

Rectified Linear Units:

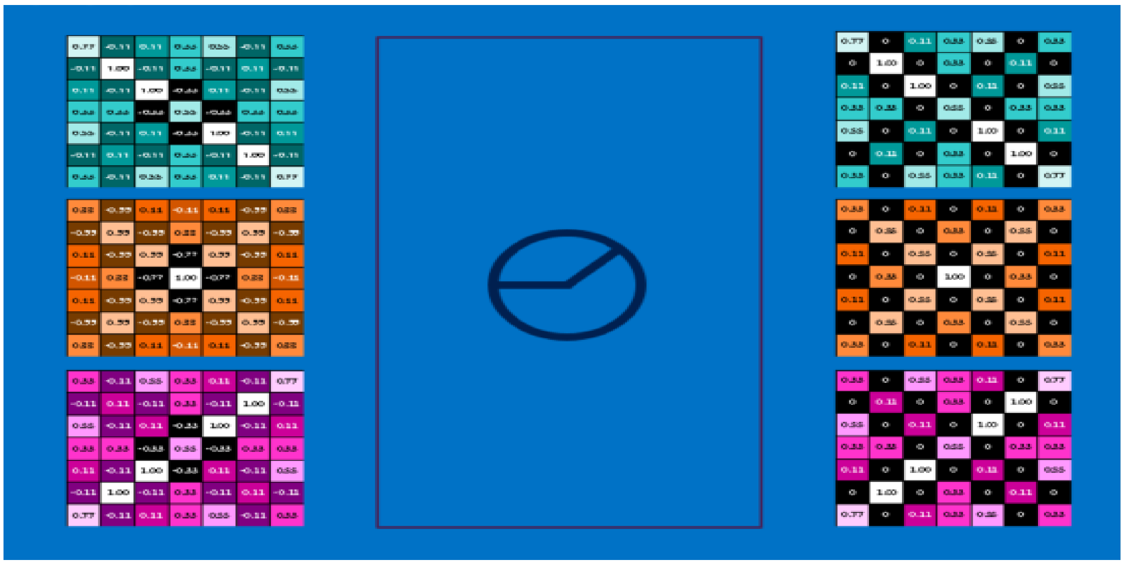
A small but important player in this process is the Rectified Linear Unit or ReLU. It's math is also very simple—wherever a negative number occurs, swap it out for a 0.



0.77	-0.11	0.11	0.33	0.55	-0.11	0.33
-0.11	1.00	-0.11	0.33	-0.11	0.11	-0.11
0.11	-0.11	1.00	-0.33	0.11	-0.11	0.55
0.33	0.33	-0.33	0.55	-0.33	0.33	0.33
0.55	-0.11	0.11	-0.33	1.00	-0.11	0.11
-0.11	0.11	-0.11	0.33	-0.11	1.00	-0.11
0.33	-0.11	0.55	0.33	0.11	-0.11	0.77

Figure 6: Rectified Linear Units.

This helps the CNN stay mathematically healthy by keeping learned values from getting stuck near 0 or blowing up toward infinity. It's the axle grease of CNNs—not particularly glamorous, but without it they don't get very far.



The output of a ReLU layer is the same size as whatever is put into it, just with all the negative values removed.

Deep learning:

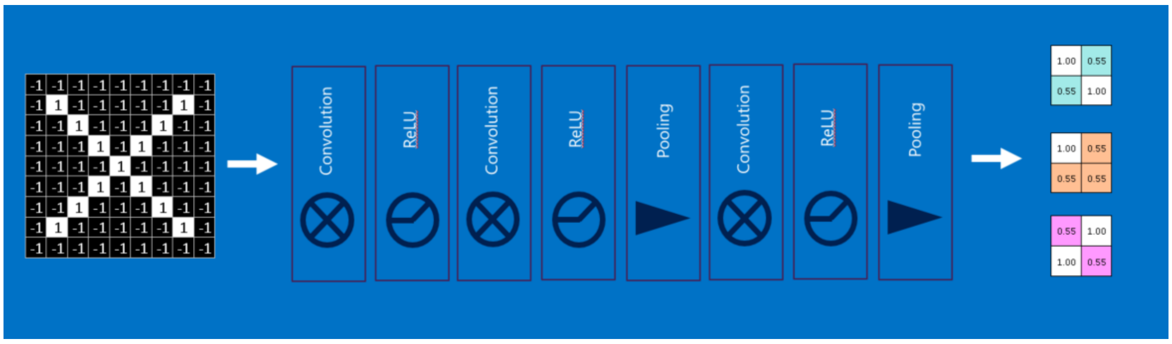
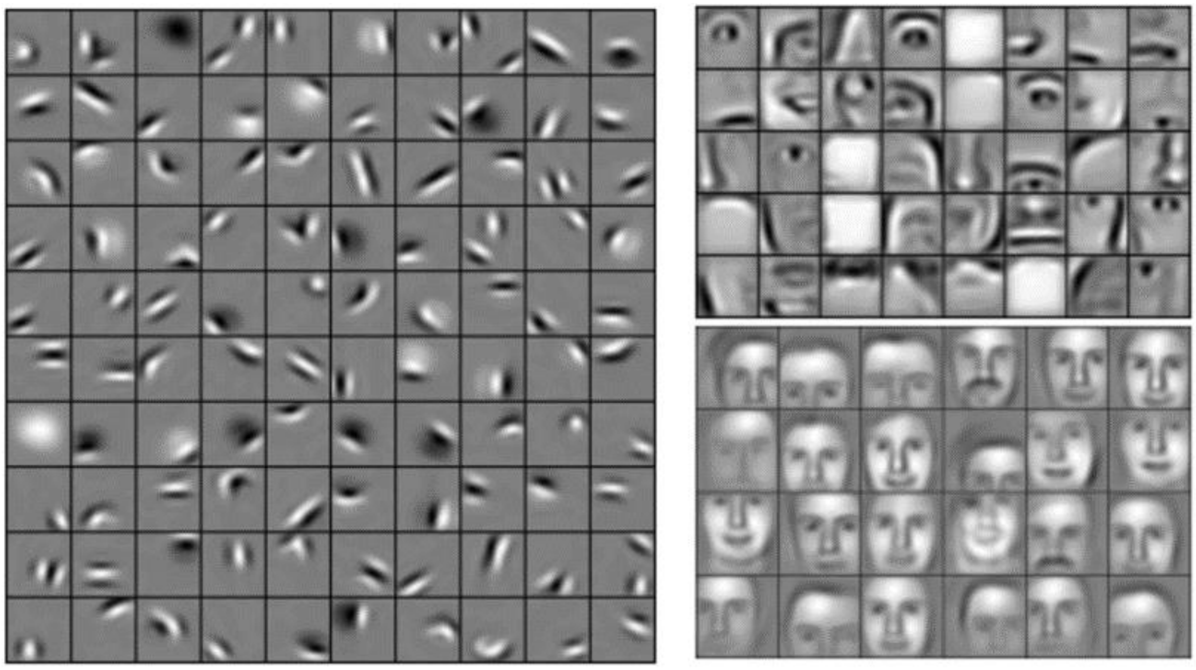


Figure 7: Deep learning.

You’ve probably noticed that the input to each layer (two-dimensional arrays) looks a lot like the output (two-dimensional arrays). Because of this, we can stack them like Lego bricks. Raw images get filtered, rectified and pooled to create a set of shrunken, feature-filtered images. These can be filtered and shrunken again and again. Each time, the features become larger and more complex, and the images become more compact. This lets lower layers represent simple aspects of the image, such as edges and bright spots. Higher layers can represent increasingly sophisticated aspects of the image, such as shapes and patterns. These tend to be readily recognizable. For instance, in a CNN trained on human faces, the highest layers represent patterns that are clearly face-like.



Fully connected layers:

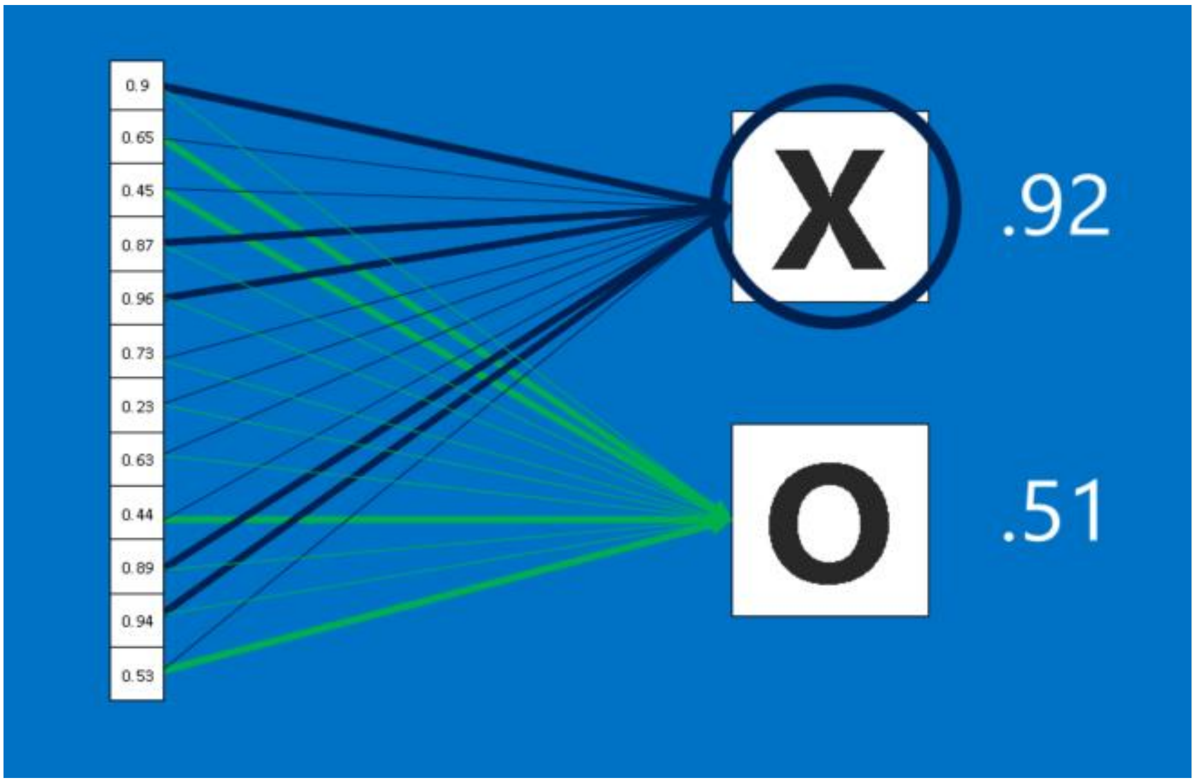
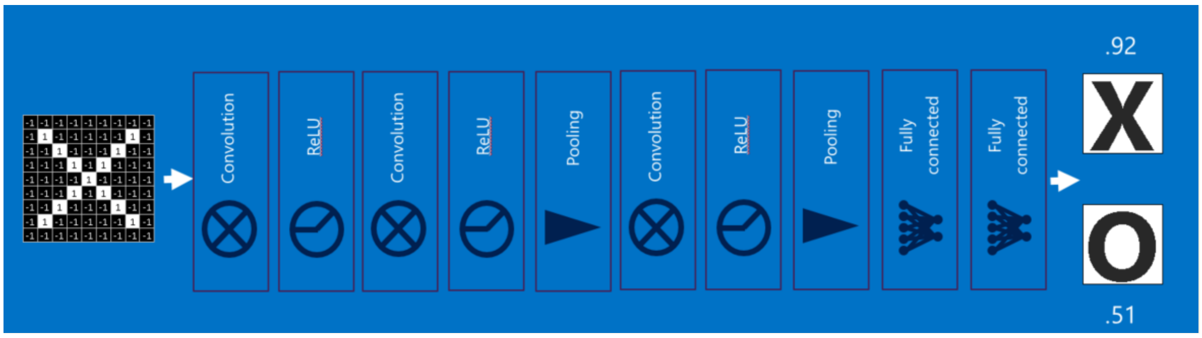


Figure 8: Fully Connected Layers.

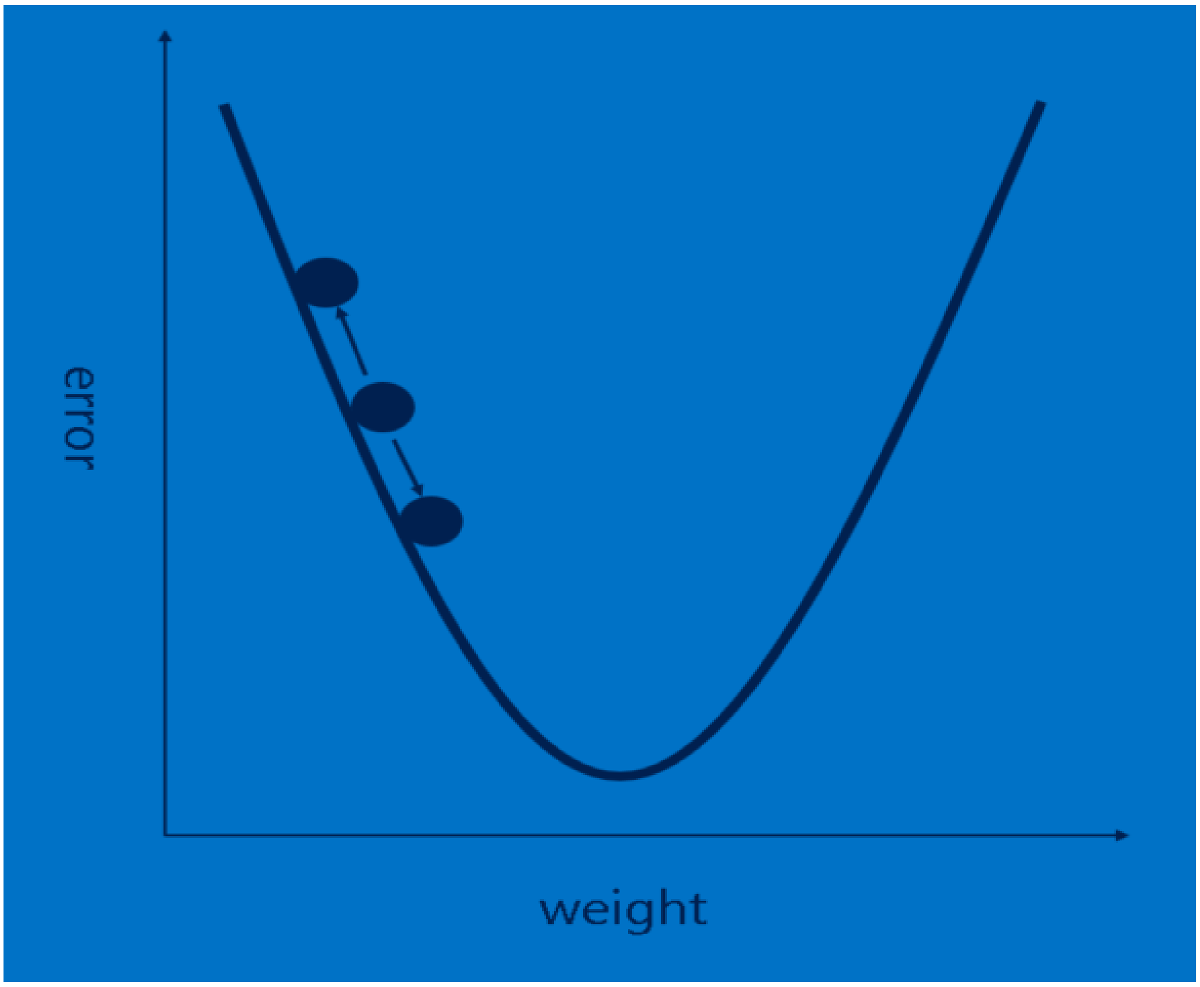
CNNs have one more arrow in their quiver. Fully connected layers take the high-level filtered images and translate them into votes. In our case, we only have to decide between two categories, X and O. Fully connected layers are the primary building block of traditional neural networks. Instead of treating inputs as a two-dimensional array, they are treated as a single list and all treated identically. Every value gets its own vote on whether the current image is an X or and O. However, the process isn’t entirely democratic. Some values are much better than others at knowing when the image is an X, and some are particularly good at knowing when the image is an O. These get larger votes than the others. These votes are expressed as weights, or connection strengths, between each value and each category. When a new image is presented to the CNN, it percolates through the lower layers until it reaches the fully connected layer at the end. Then an election is held. The answer with the most votes wins and is declared the category of the input.

Fully connected layers, like the rest, can be stacked because their outputs (a list of votes) look a whole lot like their inputs (a list of values).



In practice, several fully connected layers are often stacked together, with each intermediate layer voting on phantom "hidden" categories. In effect, each additional layer lets the network learn ever more sophisticated combinations of features that help it make better decisions.

Backpropagation:



Our story is filling in nicely, but it still has a huge hole—Where do features come from? and How do we find the weights in our fully connected layers? If these all had to be chosen by hand, CNNs would be a good deal less popular than they are. Luckily, a bit of machine learning magic called backpropagation does this work for us.

To make use of backpropagation, we need a collection of images that we already know the answer for. This means that some patient soul flipped through thousands of images and assigned them a label of X or O. We use these with an untrained CNN, which means that every pixel of every feature and every weight in every fully connected layer is set to a random value. Then we start feeding images through it, one after other.

Each image the CNN processes results in a vote. The amount of wrongness in the vote, the error, tells us how good our features and weights are. The features and weights can then be adjusted to make the error less. Each value is adjusted a little higher and a little lower, and the new error computed each time. Whichever adjustment makes the error less is kept. After doing this for every feature pixel in every convolutional layer and every weight in every fully connected layer, the new weights give an answer that works slightly better for that image. This is then repeated with each subsequent image in the set of labeled images. Quirks that occur in a single image are quickly forgotten, but patterns that occur in lots of images get baked into the features and connection weights. If you have enough labeled images, these values stabilize to a set that works pretty well across a wide variety of cases. As is probably apparent, backpropagation is another expensive computing step, and another motivator for specialized computing hardware.

4.4 Limitations:

Despite the power and resource complexity of CNNs, they provide in-depth results. At the root of it all, it is just recognizing patterns and details that are so minute and inconspicuous that it goes unnoticed to the human eye. But when it comes to understanding the contents of an image it fails.

Let’s take a look at this example. When we pass the below image to a CNN it detects a person in their mid-30s and a child probably around 10 years. But when we look at the same image we start thinking of multiple different scenarios. Maybe it’s a father and son day out, a picnic or maybe they are camping. Maybe it is a school ground and the child scored a goal and his dad is happy so he lifts him.

These limitations are more than evident when it comes to practical applications. For example, CNN’s were widely used to moderate content on social media. But despite the vast resources of images and videos that they were trained on it still isn’t able to completely block and remove inappropriate content. As it turns out it flagged a 30,000 year statue with nudity on

Facebook. Several studies have shown that CNNs trained on ImageNet and other popular datasets fail to detect objects when they see them under different lighting conditions and from new angles. Does this mean that CNNs are useless? Despite the limits of convolutional neural networks, however, there’s no denying that they have caused a revolution in artificial intelligence. Today, CNN’s are used in many computer vision applications such as facial recognition, image search, and editing, augmented reality, and more. As advances in convolutional neural networks show, our achievements are remarkable and useful, but we are still very far from replicating the key components of human intelligence.

Chapter 5

Overview of YOLO

5.1 Introduction:

People glancing at an image, can instantly recognize what the objects are and where they are located within the image. The ability to detect objects fast combined with the knowledge of a person helps to make an accurate judgment about the nature of the object. A system that simulates the ability of the human visual system to detect objects is something that scientists are researching on. Fast and accurate are the two prerequisites for which an object detection algorithm is examined.

Object detection is one of the classical problems in computer vision. It not only classifies the object in image but also localizes that object. In previous decades, the methods used to address this problem consisted of two stages: (1) extract different areas in the image using sliding windows of different sizes and (2) apply the classification problem to determine what class the objects belong to. These approaches have the disadvantage of demanding a large amount of computation and being broken down into multiple stages. That makes the system difficult to be optimized in terms of Speed.

Over the past decades, convolutional neural networks (CNNs) have been certified to be useful models for processing a wide range of visual tasks, and we have witnessed the rapid development of general object detectors. The commonly used target detection framework is divided into two branches, two-stage detectors and one-stage detectors. Typical algorithms of two-stage detectors include faster RCNN, RCNN, PANet, SPPNet, and Mask R-CNN. The second is one-stage detectors, derived from SSD, YOLOv4 to YOLOv5, and RetinaNet. The former has higher detection accuracy, but its detection speed is slower, while the latter improves the detection speed and maintains performance. At the same time, the design of face detector gain has achieved the state-of-the-art (SOTA) architecture of general object detectors. YOLO - You Only Look Once is an algorithm proposed by by Redmond et. al in a research article published at the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) as a conference paper, winning OpenCV People’s Choice Award.Compared to the approach taken by object detection algorithms before YOLO, which repurpose classifiers to perform detection, YOLO proposes the use of an end-to-end neural network that makes predictions of bounding boxes and class probabilities all at once.

YOLO vs. other detectorsIn addition to increased accuracy in predictions and a better Intersection over Union in bounding boxes (compared to real-time object detectors), YOLO has the inherent advantage of speed.YOLO is a much faster algorithm than its counterparts, running at as high as 45 FPS.Here's how YOLO works in practice

Following a fundamentally different approach to object detection, YOLO achieves state-of-the-art results beating other real-time object detection algorithms by a large margin. We consider face detectors as a special task of general objector detection. General target detection is aimed at multiple categories, while face detection is a dichotomous problem that only detects the face category. YOLOv5 which has been verified for its superior performance in general target detection tasks. To resolve the challenge of multiscale, small faces, low-light, and dense scenes.

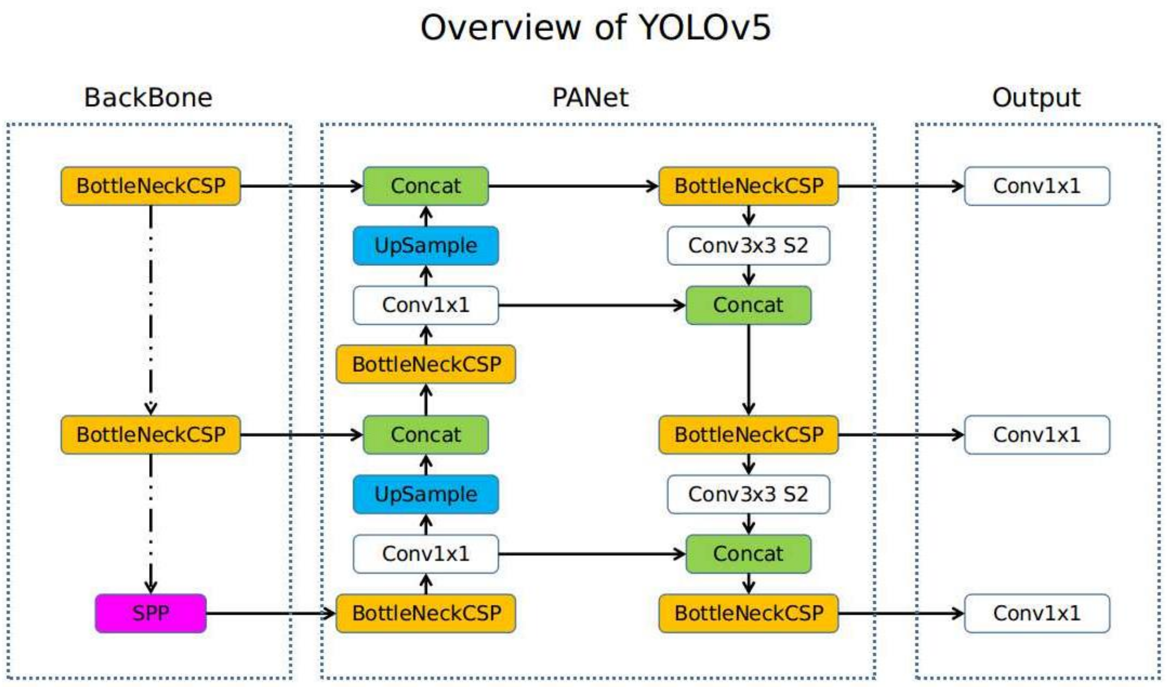


Figure 9: Yolo Structure.

YOLOV5 using CSPDarknet as a network of feature extraction, target information is extracted from the input image. The combination of CSP and Darknet formed the CSPDarknet. Figure 2 shows the structure of CSPDarknet. For the input tensor, CSP divides it into two parts in the channel, one part is convoluted once, the other part is convolution residuals multiple times. The tensor is obtained by multiple convolution-residual operations, and the tensor obtained by one convolution of the previous part is spliced in channel dimensions. CSP makes the output graph retain more network gradient information and maintains the performance of the network while reducing the computational effort. In the operation, the features of the previous stage can be used as the input of the next stage for up-sampling or down sampling, and at the same time, the CONCAT with the feature map of the same size in the main part. This pyramid structure makes the high-level feature map integrate the accurate position information of the low level [30] and improves the accuracy of regression.

5.2 Working principle of YOLO:

YOLO algorithm employs convolutional neural networks (CNN) to detect objects in real-time. As the name suggests, the algorithm requires only a single forward propagation through a neural network to detect objects.This means that prediction in the entire image is done in a single algorithm run. The CNN is used to predict various class probabilities and bounding boxes simultaneouslyIn a real real-life scenario, we need to go beyond locating just one object but rather multiple objects in one image. For example, a self-driving car has to find the location of other cars, traffic lights, signs, humans and take appropriate action based on this information.

In the case of bounding boxes, there are also some situations where we want to find the exact boundaries of our objects. This process is called instance segmentation, but this is a topic for another post.

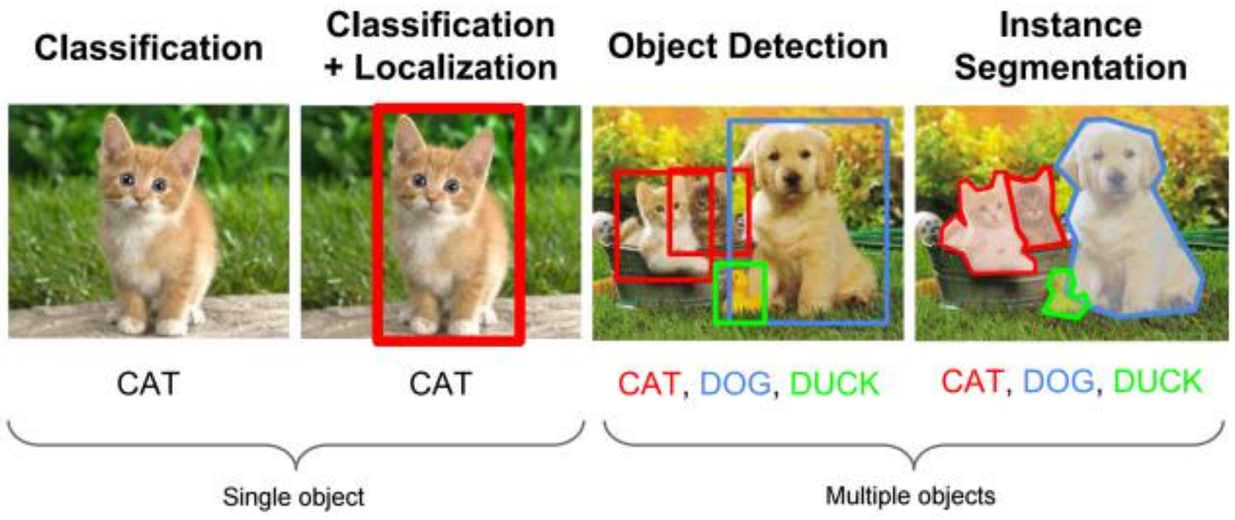


Figure 10: Image classification.

YOLO algorithm works using the following three techniques:

- Residual blocks
- Bounding box regression
- Intersection Over Union (IOU)

Residual blocks:

First, the image is divided into various grids. Each grid has a dimension of S x S. The following image shows how an input image is divided into grids.

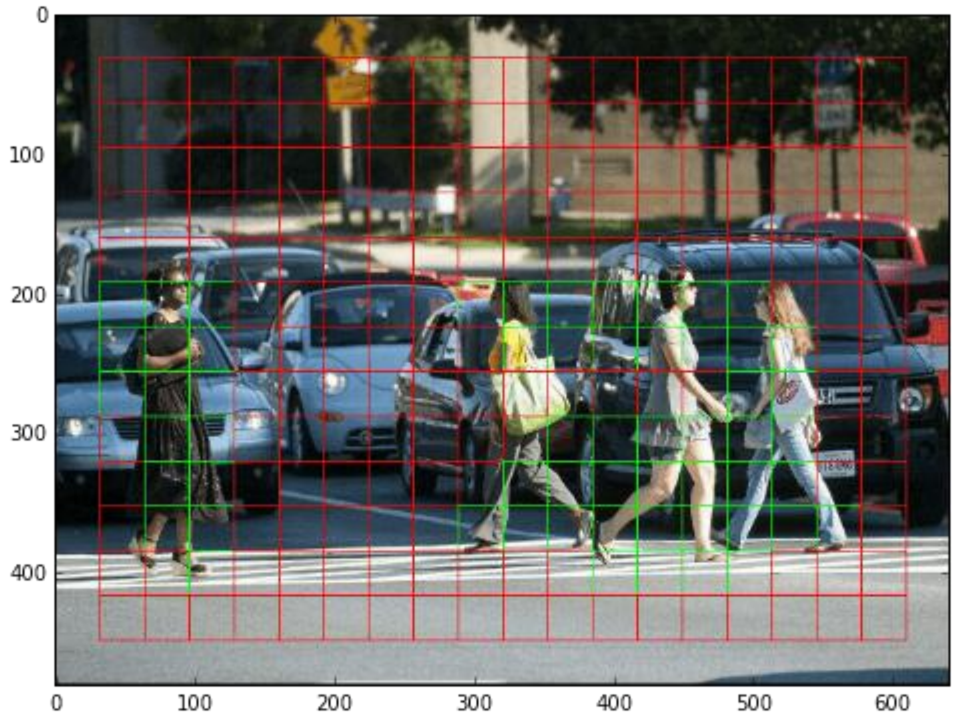


Figure 11: Image classification.

In the image above, there are many grid cells of equal dimension. Every grid cell will detect objects that appear within them. For example, if an object center appears within a certain grid cell, then this cell will be responsible for detecting it.

Bounding box regression:

A bounding box is an outline that highlights an object in an image. Every bounding box in the image consists of the following attributes:

- Width (bw)
- Height (bh)
- Class (for example, person, car, traffic light, etc.)- This is represented by the letter c.
- Bounding box center (bx,by)

The following image shows an example of a bounding box. The bounding box has been represented by a yellow outline.

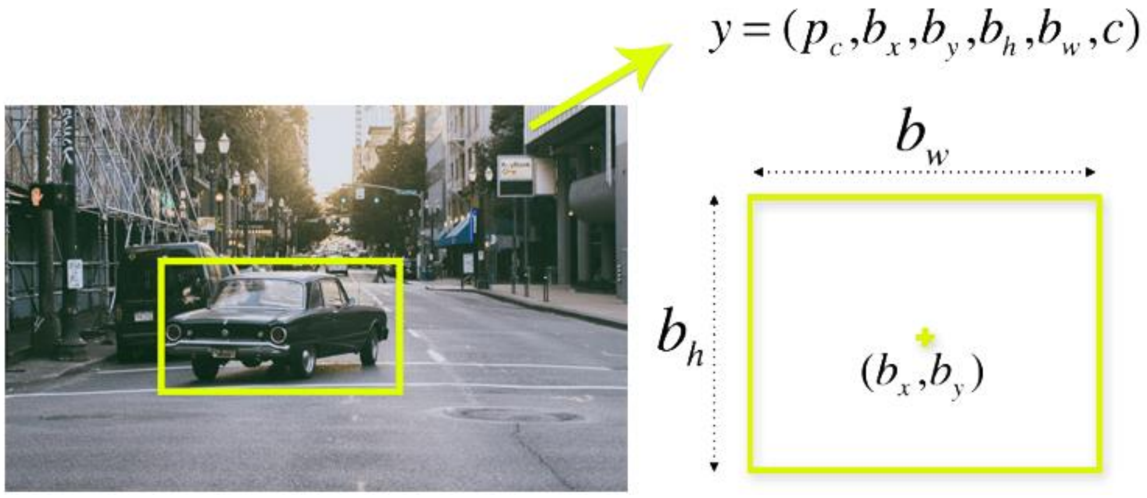


Figure 12: Image classification.

YOLO uses a single bounding box regression to predict the height, width, center, and class of objects. In the image above, represents the probability of an object appearing in the bounding box.

Intersection over union (IOU):

Here’s some food for thought – how can we decide whether the predicted bounding box is giving us a good outcome (or a bad one)? This is where Intersection over Union comes into the picture. It calculates the intersection over union of the actual bounding box and the predicted bonding box. Consider the actual and predicted bounding boxes for a car as shown below:

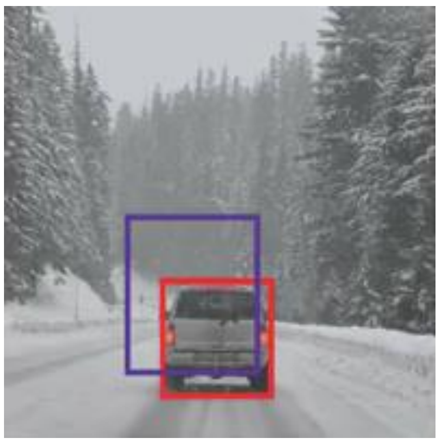


Figure 13: Bounding boxes.

Here, the red box is the actual bounding box and the blue box is the predicted one. How can we decide whether it is a good prediction or not? IoU, or Intersection over Union, will calculate the area of the intersection over union of these two boxes. That area will be:

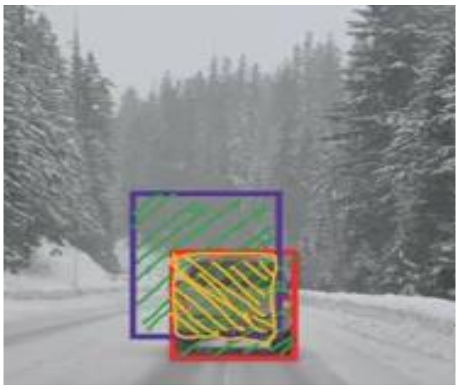


Figure 14: Bounding boxes.

IoU = Area of the intersection / Area of the union, i.e.
IoU = Area of yellow box / Area of green box

If IoU is greater than 0.5, we can say that the prediction is good enough. 0.5 is an arbitrary threshold we have taken here, but it can be changed according to your specific problem. Intuitively, the more you increase the threshold, the better the predictions become. There is one more technique that can improve the output of YOLO significantly – Non-Max Suppression. One of the most common problems with object detection algorithms is that rather than detecting an object just once, they might detect it multiple times. Consider the below image:

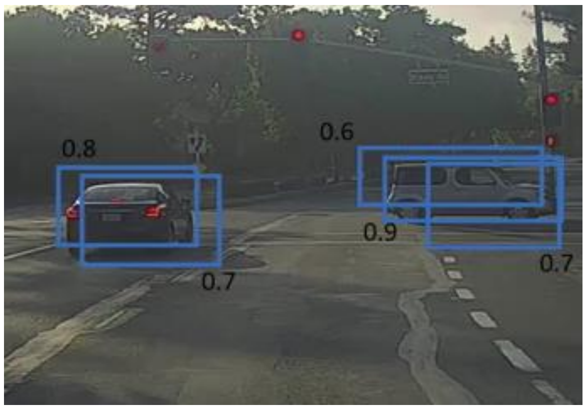


Figure 15: Bounding boxes.

Here, the cars are identified more than once. The Non-Max Suppression technique cleans up this up so that we get only a single detection per object. Let’s see how this approach works.

1. It first looks at the probabilities associated with each detection and takes the largest one. In the above image, 0.9 is the highest probability, so the box with 0.9 probability will beselected first:

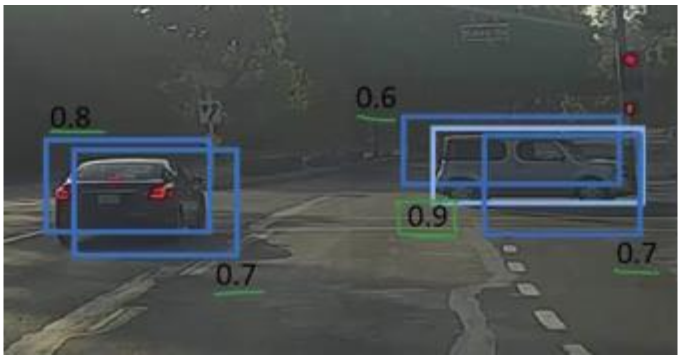


Figure 16: Bounding boxes.

2. Now, it looks at all the other boxes in the image. The boxes which have high IoU with the current box are suppressed. So, the boxes with 0.6 and 0.7 probabilities will be suppressed in our example:

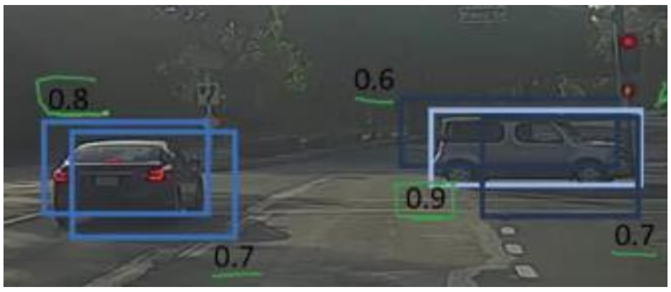
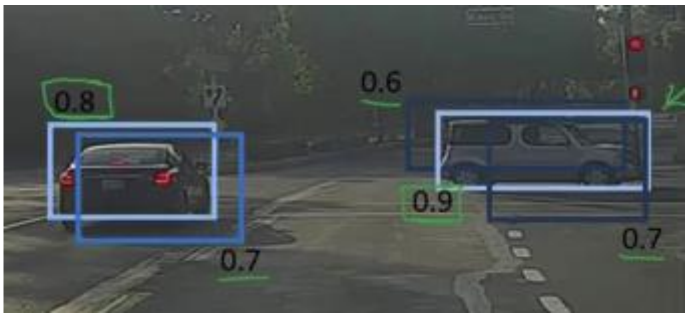


Figure 17: Bounding boxes.

3. After the boxes have been suppressed, it selects the next box from all the boxes with the highest probability, which is 0.8 in our case:



4. Again it will look at the IoU of this box with the remaining boxes and compress the boxes with a high IoU:

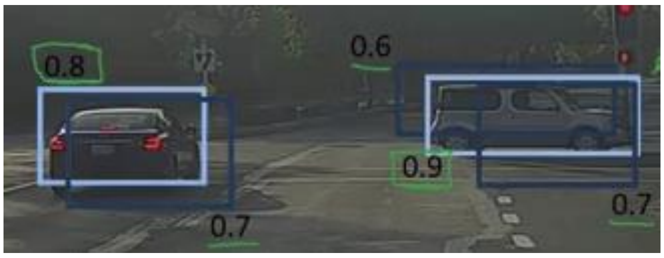


Figure 18: Bounding boxes.

5. We repeat these steps until all the boxes have either been selected or compressed and we get the final bounding boxes:

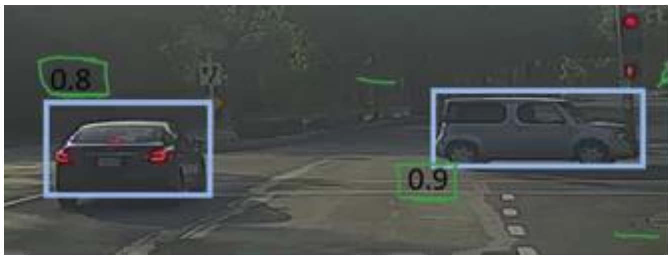


Figure 19: Bounding boxes.

This is what Non-Max Suppression is all about. We are taking the boxes with maximum probability and suppressing the close-by boxes with non-max probabilities. Let's quickly summarize the points which we've seen in this section about the Non-Max suppression algorithm:

1. Discard all the boxes having probabilities less than or equal to a pre-defined threshold (say, 0.5)
2. For the remaining boxes:
    1. Pick the box with the highest probability and take that as the output prediction.
    2. Discard any other box which has IoU greater than the threshold with the output box from the above step.
3. Repeat step 2 until all the boxes are either taken as the output prediction or discarded There is another method we can use to improve the perform of a YOLO algorithm – let's check it out!

Anchor Boxes:

We have seen that each grid can only identify one object. But what if there are multiple objects in a single grid? That can so often be the case in reality. And that leads us to the concept of anchor boxes. Consider the following image, divided into a 3 X 3 grid:

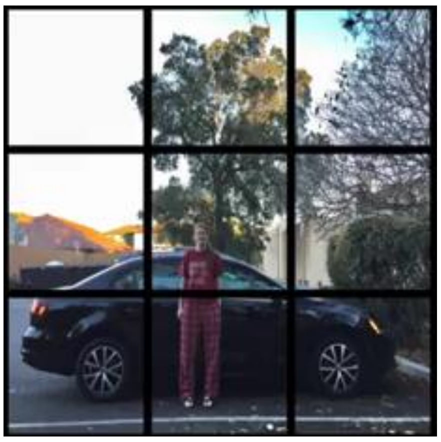


Figure 20: Anchor Boxes.

Remember how we assigned an object to a grid? We took the midpoint of the object and based on its location, assigned the object to the corresponding grid. In the above example, the midpoint of both the objects lies in the same grid. This is how the actual bounding boxes for the objects will be:

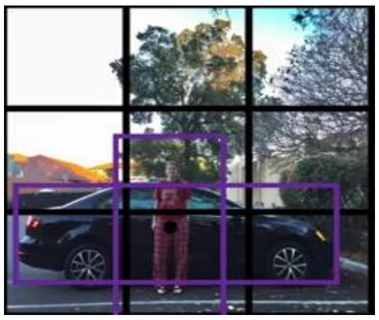


Figure 21: Anchor Boxes.

We will only be getting one of the two boxes, either for the car or for the person. But if we use anchor boxes, we might be able to output both boxes! How do we go about doing this? First, we pre-define two different shapes called anchor boxes or anchor box shapes. Now, for each grid, instead of having one output, we will have two outputs. We can always increase the number of anchor boxes as well. I have taken two here to make the concept easy to understand:

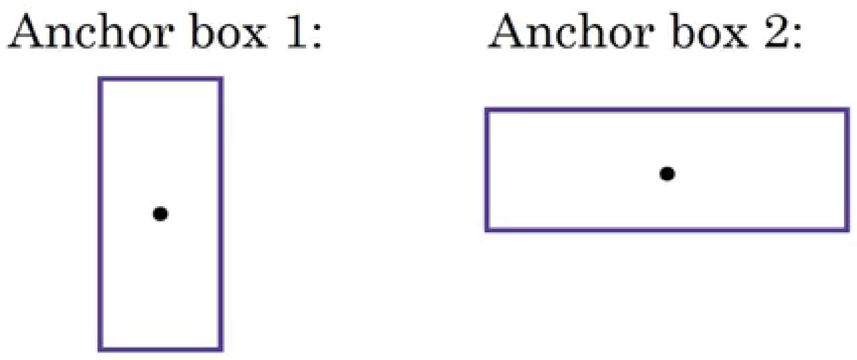
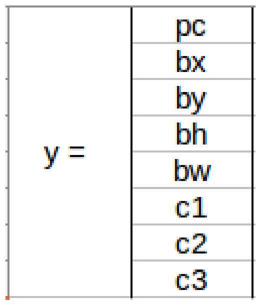


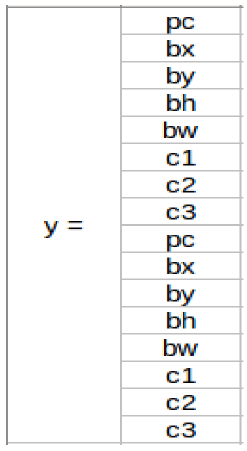
Figure 22: Anchor Boxes.

This is how the y label for YOLO without anchor boxes looks like:



y =	pc
	bx
	by
	bh
	bw
	c1
	c2
	c3

What do you think the y label will be if we have 2 anchor boxes? I want you to take a moment to ponder this before reading further. Got it? The y label will be



y =	pc
	bx
	by
	bh
	bw
	c1
	c2
	c3
	pc
	bx
	by
	bh
	bw
	c1
	c2
	c3

The first 8 rows belong to anchor box 1 and the remaining 8 belongs to anchor box 2. The objects are assigned to the anchor boxes based on the similarity of the bounding boxes and the anchor box shape. Since the shape of anchor box 1 is similar to the bounding box for the person, the latter will be assigned to anchor box 1 and the car will be assigned to anchor box

2. The output in this case, instead of 3 X 3 X 8 (using a 3 X 3 grid and 3 classes), will be 3 X 3 X 16 (since we are using 2 anchors).

So, for each grid, we can detect two or more objects based on the number of anchors. Let's combine all the ideas we have covered so far and integrate them into the YOLO framework.

Combination of the three techniques:

The following image shows how the three techniques are applied to produce the final detection results.

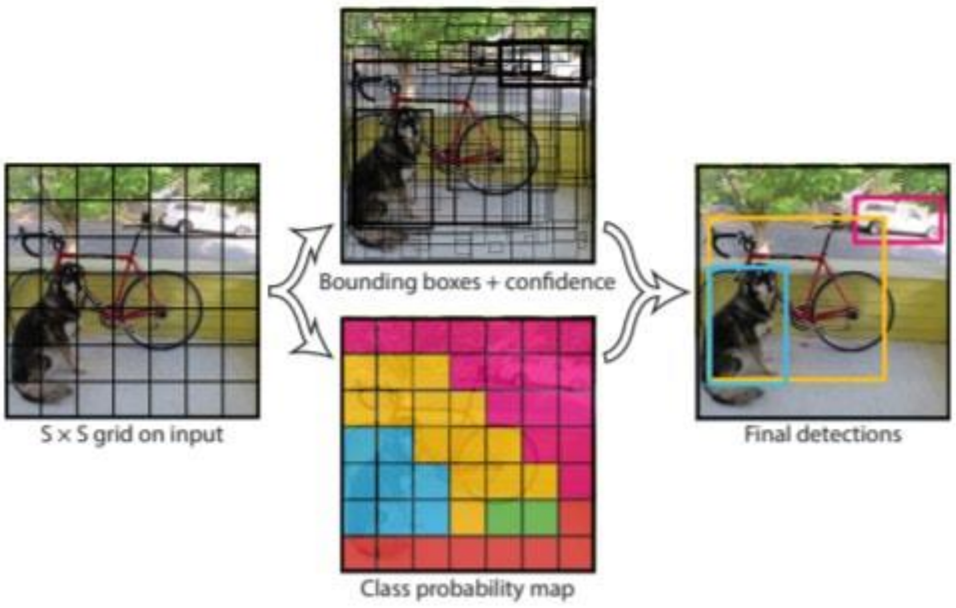


Figure 23: Combination of the three techniques

First, the image is divided into grid cells. Each grid cell forecasts B bounding boxes and provides their confidence scores. The cells predict the class probabilities to establish the class of each object.
For example, we can notice at least three classes of objects: a car, a dog, and a bicycle. All the predictions are made simultaneously using a single convolutional neural network.

Intersection over union ensures that the predicted bounding boxes are equal to the real boxes of the objects. This phenomenon eliminates unnecessary bounding boxes that do not meet the characteristics of the objects (like height and width). The final detection will consist of unique bounding boxes that fit the objects perfectly.

For example, the car is surrounded by the pink bounding box while the bicycle is surrounded by the yellow bounding box. The dog has been highlighted using the blue bounding box.

5.3 YOLO Architecture:

Inspired by the GoogleNet architecture, YOLO's architecture has a total of 24 convolutional layers with 2 fully connected layers at the end. YOLO uses features from the entire image and predicts bounding boxes simultaneously. The image is divided into S X S grid and each gird produces B bounding boxes and their confidence scores. These confidence scores reflect how confident the model is that the box contains an object and also how accurate it thinks the box is that it predicts. These confidence scores reflect how confident the model is that the box contains an object and also how accurate it thinks the box is that it predicts. Each bounding box consists of 5 predictions: x, y, w, h, and confidence. The (x, y) coordinates represent the center of the box relative to the bounds of the grid cell. The width and height are predicted relative to the whole image.

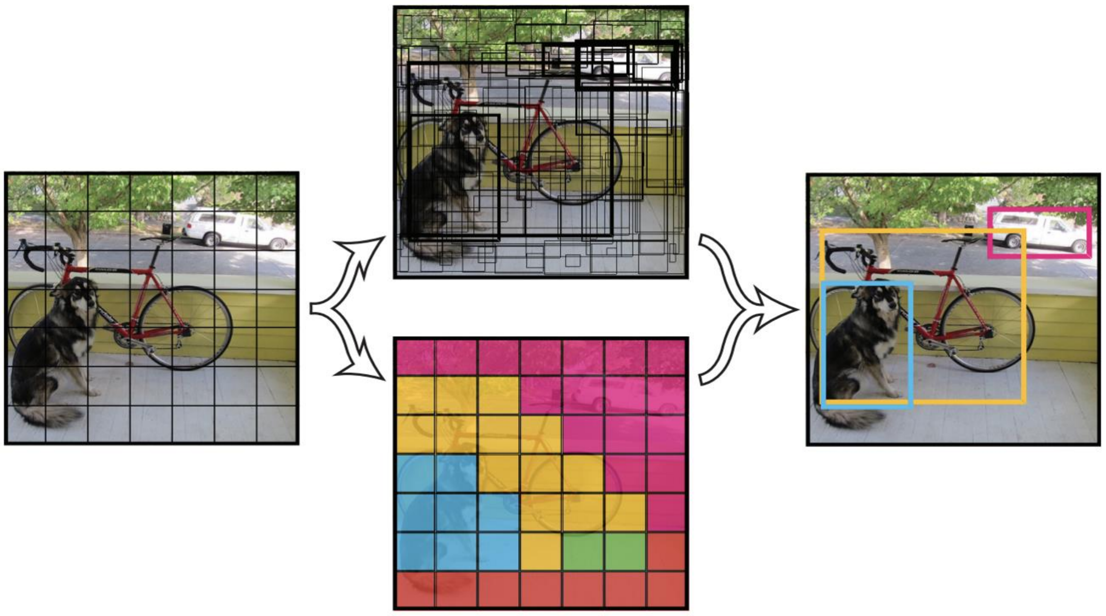


Figure 24: YOLO Architecture.

Each image is divided into boxes of equal size and the bounding boxes are drawn with each line width indicating the confidence scores.

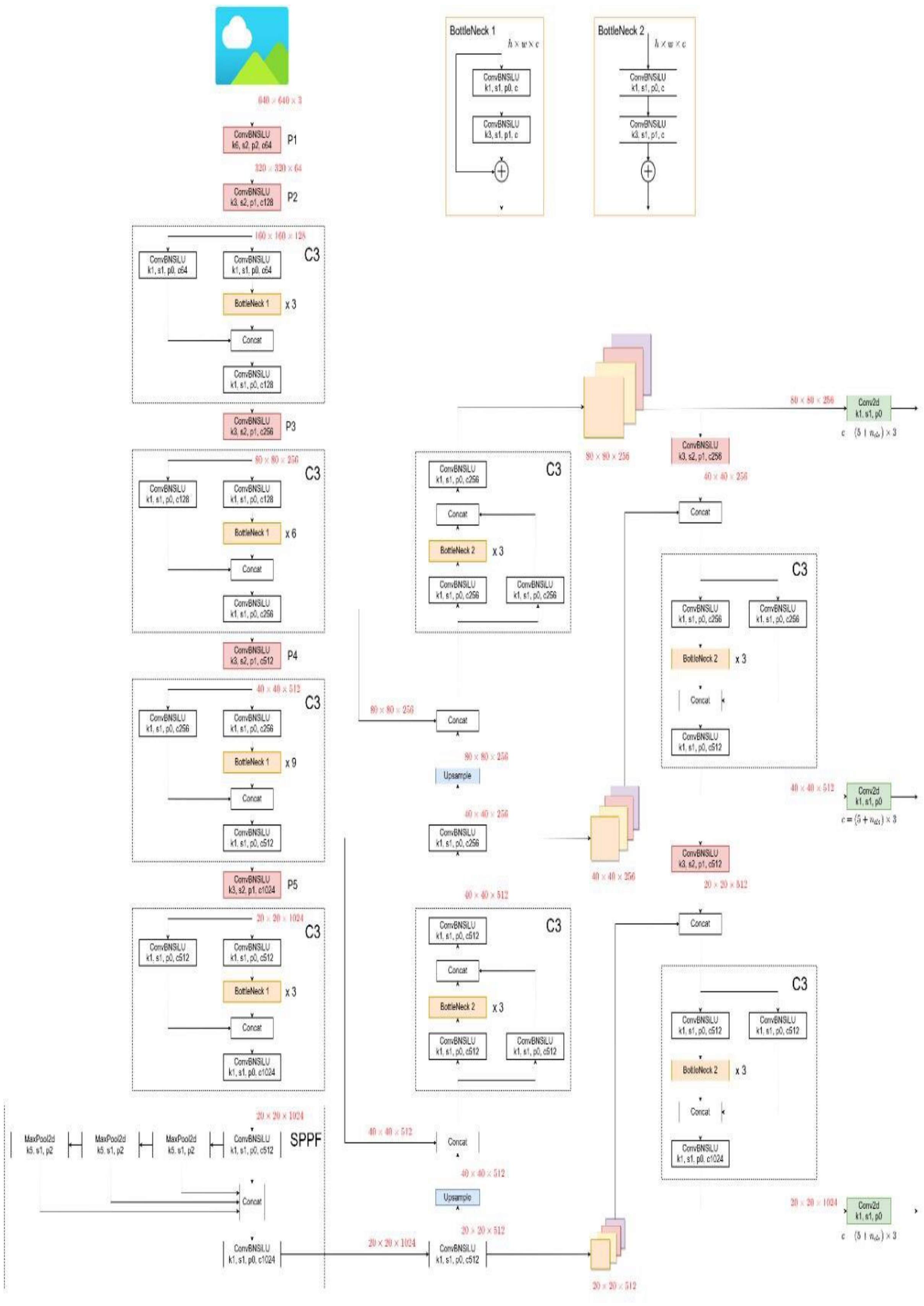
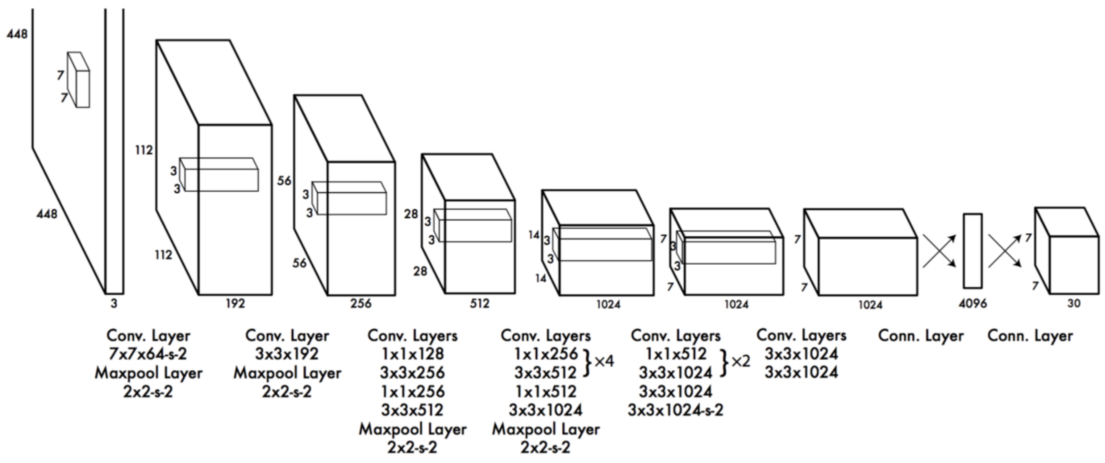


Figure 25: YOLO Architecture.

YOLO model was evaluated on Pascal VOC detection dataset. The initial convolutional layers of the network extract feature from the image while the fully connected layers predict the output probabilities and coordinates. It has 24 convolutional layers followed by 2 fully connected layers.



448
448
7
7
3
112
112
3
3
192
56
56
3
3
256
28
28
3
3
512
14
14
3
3
1024
7
7
3
3
1024
7
7
1024
4096
7
7
30
Conv. Layer 7x7x64-s-2 Maxpool Layer 2x2-s-2
Conv. Layer 3x3x192 Maxpool Layer 2x2-s-2
Conv. Layers 1x1x128 3x3x256 1x1x256 3x3x512 Maxpool Layer 2x2-s-2
Conv. Layers 1x1x256 3x3x512 }×4 1x1x512 3x3x1024 Maxpool Layer 2x2-s-2
Conv. Layers 1x1x512 3x3x1024 }×2 3x3x1024 3x3x1024-s-2
Conv. Layers 3x3x1024 3x3x1024
Conn. Layer
Conn. Layer

Figure 26: YOLO Layers.

5.4 YOLO limitations:

YOLO imposes strong spatial constraints on bounding box predictions since each grid cell only predicts two boxes and can only have one class. This spatial constraint limits the number of nearby objects that our model can predict. Our model struggles with small objects that appear in groups, such as flocks of birds. Since our model learns to predict bounding boxes from data, it struggles to generalize to objects in new or unusual aspect ratios or configurations. Our model also uses relatively coarse features for predicting bounding boxes since our architecture has multiple down sampling layers from the input image. Finally, while we train on a loss function that approxi-mates detection performance, our loss function treats errors the same in small bounding boxes versus large bounding boxes. A small error in a large box is generally benign but a small error in a small box has a much greater effect on IOU.Our main source of error is incorrect localizations.YOLO struggles to detect and segregate small objects in images that appear in groups, as each grid is constrained to detect only a single object. Small objects that naturally come in groups, such as a line of ants, are therefore hard for YOLO to detect and localize. YOLO is also characterized by lower accuracy when compared to much slower object detection algorithms like Fast RCNN. Now, before we deep dive into more details about the YOLO architecture and methodology, let's go over some of the important terminology.

Chapter 6

SLAM

6.1 Introduction:

The Simultaneous Localization and Mapping (SLAM) problem can be defined as a process where a robot builds a map representing its spatial environment while keeping rack of its position within the built map. Mapping is done online with no prior knowledge of the robot’s location; the built map is subsequently used by the robot for navigation. SLAM is a key component of any truly autonomous robot. Much recent research has been done tackling the computational efficiency of SLAM and the data association and landmark extraction necessary for a robust SLAM method. Environmental mapping involves creating a mathematical model of a real environment’s spatial information. SLAM extends the requirements of this mathematical model; it must also jointly represent the robot’s state and the position of extracted landmarks relative to the robot’s location. The robot’s state includes information on the robot’s position and orientation. The basic SLAM framework involves odometry, landmark prediction, landmark extraction, data association and matching, pose estimation, and map update. These processes are the backbone of every major SLAM method, and are performed in cyclic fashion. Mapping the spatial information of an environment requires spatial sensors to which SLAM algorithms can be applied. The two most popular sensor modalities used in SLAM are raw range scan sensors and feature (landmark) based sensors (whether extracted from scans or images)14. The most commonly used sensors for landmark extraction from scans are laser-based and sonar-based system. Landmark extraction from images (often referred to as visual SLAM) uses camera in a variety of configurations such as monocular configuration4, stereo vision configuration7,20 and multiple camera configuration12. Visual sensing provides information-heavy data, but with significant amounts of noise and uncertainty.

The need to use a map of the environment is twofold. First, the map is often required to support other tasks; for instance, a map can inform path planning or provide an intuitive visualization for a human operator. Second, the map allows limiting the error committed in estimating the state of the robot. In the absence of a map, dead-reckoning would quickly drift over time; on the other hand, using a map, e.g., a set of distinguishable landmarks, the robot can “reset” its localization error by re-visiting known areas (so-called loop closure). Therefore, SLAM finds applications in all scenarios in which a prior map is not available and needs to be built. In some robotics applications the location of a set of landmarks is known a priori. For instance, a robot operating on a factory floor can be provided with a manually-built map of artificial beacons in the environment. Another example is the case in which the robot has access to GPS (the GPS satellites can be considered as moving beacons at known locations). In such scenarios, SLAM may not be required if localization can be done reliably with respect to the known landmarks. The popularity of the SLAM problem is connected with the emergence of indoor applications of mobile robotics. Indoor operation rules out the use of GPS to bound the localization error; furthermore, SLAM provides an appealing alternative to user-built maps, showing that robot operation is possible in the absence of an ad hoc localization infrastructure

6.2 About SLAM:

The term SLAM is as stated an acronym for Simultaneous Localization And Mapping. It was originally developed by Hugh Durrant-Whyte and John J. Leonard [7] based on earlier work by Smith, Self and Cheeseman [6]. Durrant-Whyte and Leonard originally termed it SMAL but it was later changed to give a better impact. SLAM is concerned with the problem of building a map of an unknown environment by a mobile robot while at the same time navigating the environment using the map. SLAM consists of multiple parts; Landmark extraction, data association, state estimation, state update and landmark update. There are many ways to solve each of the smaller parts. We will be showing examples for each part. This also means that some of the parts can be replaced by a new way of doing this. As an example we will solve the landmark extraction problem in two different ways and comment on the methods. The idea is that you can use our implementation and extend it by using your own novel approach to these algorithms. We have decided to focus on a mobile robot in an indoor environment. You may choose to change some of these algorithms so that it can be for example used in a different environment. SLAM is applicable for both 2D and 3D motion. We will only be considering 2D motion. It is helpful if the reader is already familiar with the general concepts of SLAM on an introductory level, e.g. through a university level course on the subject. There are lots of great introductions to this field of research including . Also it is helpful to know a little about the Extended Kalman Filter (EKF); sources of introduction are. Background information is always helpful as it will allow you to more easily understand this tutorial but it is not strictly required to comprehend all of it.

6.3 Lidar-SLAM:

LiDAR can detect the distance of the obstacles, and it is the best sensor to construct a grid map, which represents the structure and obstacles on the robot running plane. The early SLAM research often used LiDAR as the main sensor. Extended Kalman filter (EKF) was applied to estimate the pose and of the robot, but the performance was not ideal. For some strong nonlinear systems, this method will bring more truncation errors, which may lead to inaccurate positioning and mapping. Particle filter approaches were introduced because they can effectively avoid the nonlinear problem, but it also leads to the problem of increasing the amount of calculation with the increase of particle number. In 2007, Grisetti proposed a milestone of LiDAR-SLAM method called Gmapping , based on improved Rao-Blackwellized particle filter (RBPF), it improves the positioning accuracy and reduces the computational complexity by improving the proposed distribution and adaptive re-sampling technique. As an effective alternative to probabilistic approaches, optimization-based methods are popular in recent years. In 2010, Kurt Konolige proposed such a representative method called Karto-SLAM, which uses sparse pose adjustment to solve the problem of matrix direct solution in nonlinear optimization. Hector SLAM proposed in 2011 uses the Gauss-Newton method to solve the problem of scanning matching, this method does not need odometer information, but high precision LiDAR is required. In 2016, Google put forward a notable method called Cartographer, by applying the laser loop closing to both sub-maps and global map, the accumulative error is reduced.

6.4 Visual-SLAM:

Visual-SLAM Using visual sensors to build environment map is another hot spot for robot navigation. Compared with LiDAR-SLAM, Visual-SLAM is more complex, because image carries too much information, but has difficulty in distance measurement. Estimating robot motion by matching extracted image features under different poses to build a feature map is a common method for Visual-SLAM. Mono-SLAM [6], proposed in 2007, is considered the origin of many Visual-SLAM. Extended Kalman filter (EKF) is used as the back-end to track the sparse feature points in the front-end. The uncertainty is expressed by a probability density function. From the observation model and recursive calculation, the mean and variance of the posterior probability distribution are obtained. Reference used RBPF to realize visual-SLAM. This method avoids the problem of non-linear and has high precision, but it needs a large number of particles, which increase the computational complexity. PTAM is a representative work of visual-SLAM, it proposed a simple and effective method to extract key frames, as well as a parallel framework of a real-time tracking thread and a back-end nonlinear optimization mapping thread. It is the first time to put forward the concept of separating the front and back ends, leading the structure design of many SLAM methods. ORB-SLAM [9] is considered as a milestone of visual-SLAM. By applying Oriented FAST and Rotated BRIEF (ORB) features and bag-of-words (BOW) model, this method can create a feature map of the environment in real-time stably in many cases. Loop detection and closing via BOW is an outstanding contribution of this work, it effectively prevents the cumulative error and can be quickly retrieved after the tracking lost. In recent years, different from feature-based methods, some direct methods of visual-SLAM were explored, by estimating robot motion through pixel value directly. Dense image alignment based on each pixel of the images proposed in Reference [10] can build a dense 3D map of the environment. The work in Reference built a semi-dense map by estimating the depth values of pixels with a large gradient in the image. Engel et al. proposed LSD-SLAM (Large-Scale Direct Monocular SLAM) algorithm. The core of LSD-SLAM algorithm is to apply a direct method to semi-dense monocular SLAM, which is rarely seen in the previous direct method. Forster et al. proposed SVO (Semi-Direct Monocular Visual Odometry) in 2014, called “sparse direct method”, which combines feature points with direct methods to track some key points (such as corners, etc.), and then estimates the camera motion and position according to the information around the key points. This method runs fast for Unmanned Aerial Vehicles (UAV), by adding special constraints and optimization to such applications RGB-D camera can provide both color and depth information in its view field. It is the most capable sensor for building a complete 3D scene map. Reference proposes Kinect fusion method, which uses the depth images acquired by Kinect to measure the minimum distance of each pixel in each frame, and fuses all the depth images to obtain global map information. Reference constructs error function by using photo-metric and geometric information of image pixels. Camera pose is obtained by minimizing the error function. Map problem is treated as pose graph representation. Reference is a better direct RGB-D SLAM method. This method combines the intensity error and depth error of pixels as error functions, and minimizes the cost function to obtain the optimal camera pose. This process is implemented by g2o. Entropy-based key frame extraction and closed-loop detection method are both proposed, thus greatly reducing the path error.

6.5 Integrated LiDAR Visual-based SLAM:

In this study, the sensors are configured competitively to improve the output odometry. Visual-based SLAM provides loop closure detection to increase localization accuracy. LiDAR-based SLAM provides wider range of data to increase the field of view (FOV) of the system to overcome featureless environment. Both sensors provide point cloud data with the information of a surrounding environment. Then, feature-matching algorithm will be applied to 3D point cloud to update the map and 2D point cloud will be used to deduce the odometry information. The pose information of the robot is obtained based onintegrated information. The sensor fusion flow chart is presented in Fig:

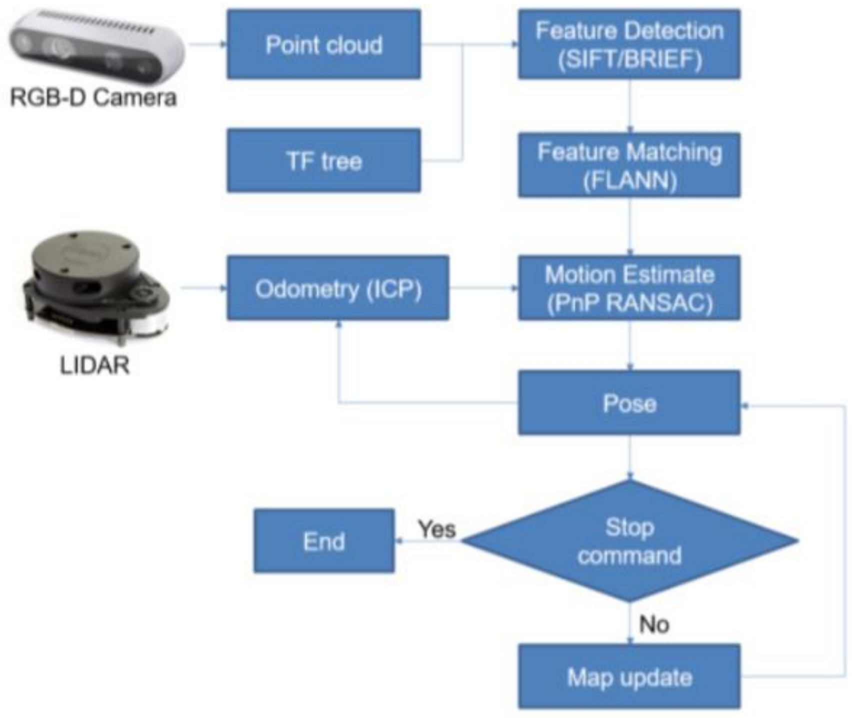


Figure 27: Hybrid SLAM.

The SIFT algorithm has four operations. Firstly, it estimates a scale space extreme based on the Difference of Gaussian (DoG). Secondly, it finds the key point localization by eliminating the low contrast points. Thirdly, a key point orientations are obtained based on local image gradient. Finally, it computes a descriptor for the local image region. For more detail, please refer to [24]. Binary Robust Independent Elementary Features (BRIEF) is another alternative method, which is applied in this study as requests less complexity than SIFT with similar matching performance. Feature Matching algorithm is implemented using Fast Approximate Nearest Neighbor Search (FLANN). Then, PnP (perspective-npoint) and RANSAC (Random Sample Consensus) are applied to enhance motion estimation. Those features will be used to compare the older frame with newer frame to deduce the robot position and update the map. Iterationately, the closest neighbor of each point in the source is found by using a search algorithm and the rigid body transformation between the target points and their closest neighbors. The entered target point cloud is then transformed using the rigid body transformation estimation and a new closest neighbor search is performed. This process is iterated until convergence. The 2D point cloud also gives information of the robot position using ICP (Iterative Closest Point) to minimize the difference between two point-clouds. In ICP algorithm, the target is fixed while the source is transformed. The data from LiDAR and 3D camera are configured competitively to improve the odometry.

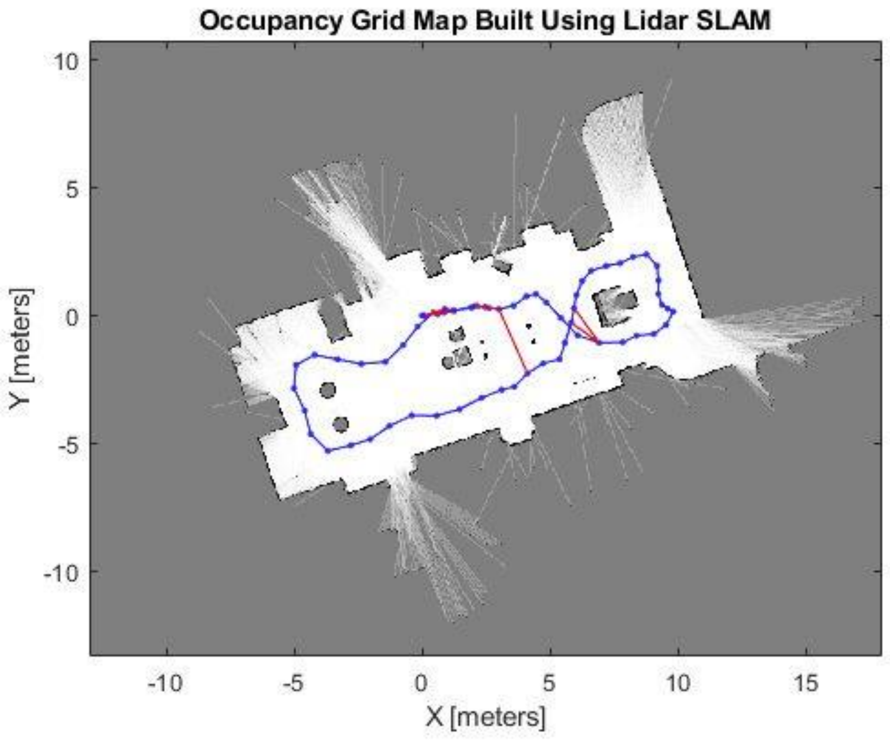


Figure 28: SLAM with 2D LiDAR

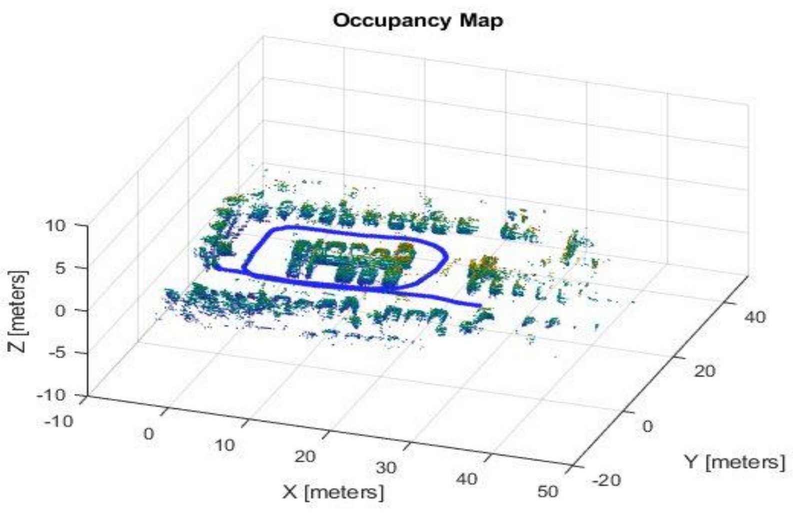


Figure 29: SLAM with 3D LiDAR

The mapping process using ROS provides a graphical user interface named as rtabmapviz, which visualizes visual odometry, output of the loop closure detector, and a point cloud that is a 3D dense map. The reconstructed map is shown in Figure 5. In this experiment, the largest linear error/angular error of the combination method is smaller than sole methods.

6.6 The SLAM Process:

The SLAM process consists of a number of steps. The goal of the process is to use the environment to update the position of the robot. Since the odometry of the robot (which gives the robots position) is often erroneous we cannot rely directly on the odometry. We can use laser scans of the environment to correct the position of the robot. This is accomplished by extracting features from the environment and reobserving when the robot moves around. An EKF (Extended Kalman Filter) is the heart of the SLAM process. It is responsible for updating where the robot thinks it is based on these features. These features are commonly called landmarks and will be explained along with the EKF in the next couple of chapters. The EKF keeps track of an estimate of the uncertainty in the robot's position and also the uncertainty in these landmarks it has seen in the environment.
An outline of the SLAM process is given below.

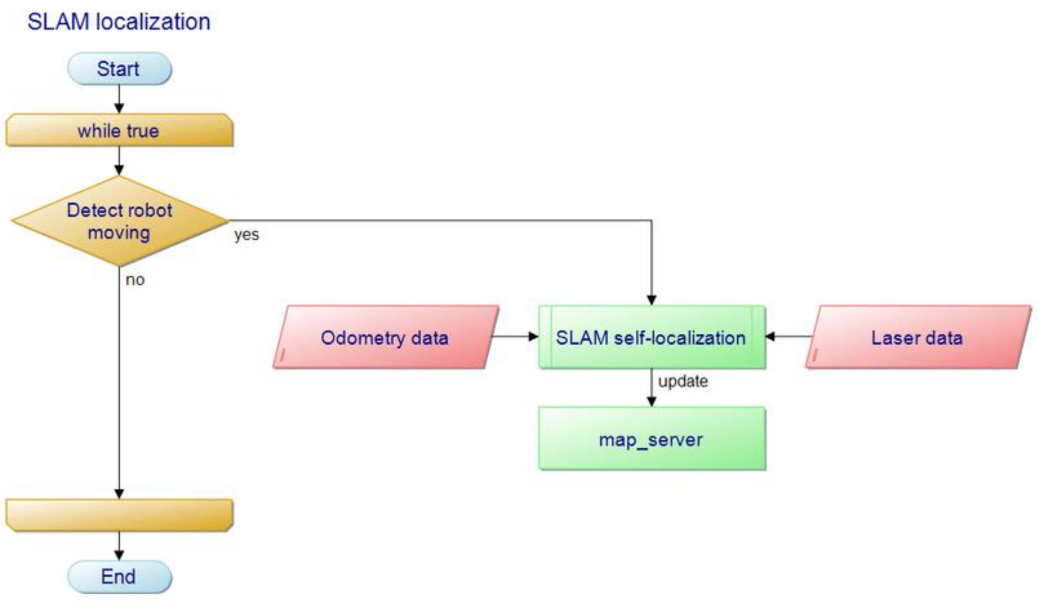


Figure 30: SLAM process.

When the odometry changes because the robot moves the uncertainty pertaining to the robot's new position is updated in the EKF using Odometry update. Landmarks are then extracted from the environment from the robot's new position. The robot then attempts to associate these landmarks to observations of landmarks it previously has seen. Re-observed landmarks are then used to update the robots position in the EKF. Landmarks which have not previously been seen are added to the EKF as new observations so they can be re-observed later. All these steps will be explained in the next chapters in a very practical fashion relative to how our ER1 robot was implemented. It should be noted that at any point in these steps the EKF will have an estimate of the robot’s current position.

6.7 Common Challenges with SLAM:

Although SLAM is used for some practical applications, several technical challenges prevent more general-purpose adoption. Each has a countermeasure that can help overcome the obstacle.

6.7.1 Localization errors:

SLAM estimates sequential movement, which include some margin of error. The error accumulates over time, causing substantial deviation from actual values. It can also cause map data to collapse or distort, making subsequent searches difficult. Let’s take an example of driving around a square-shaped passage. As the error accumulates, robot’s starting and ending point no longer match up. This is called a loop closure problem. Pose estimation errors like these are unavoidable. It is important to detect loop closures and determine how to correct or cancel out the accumulated error.

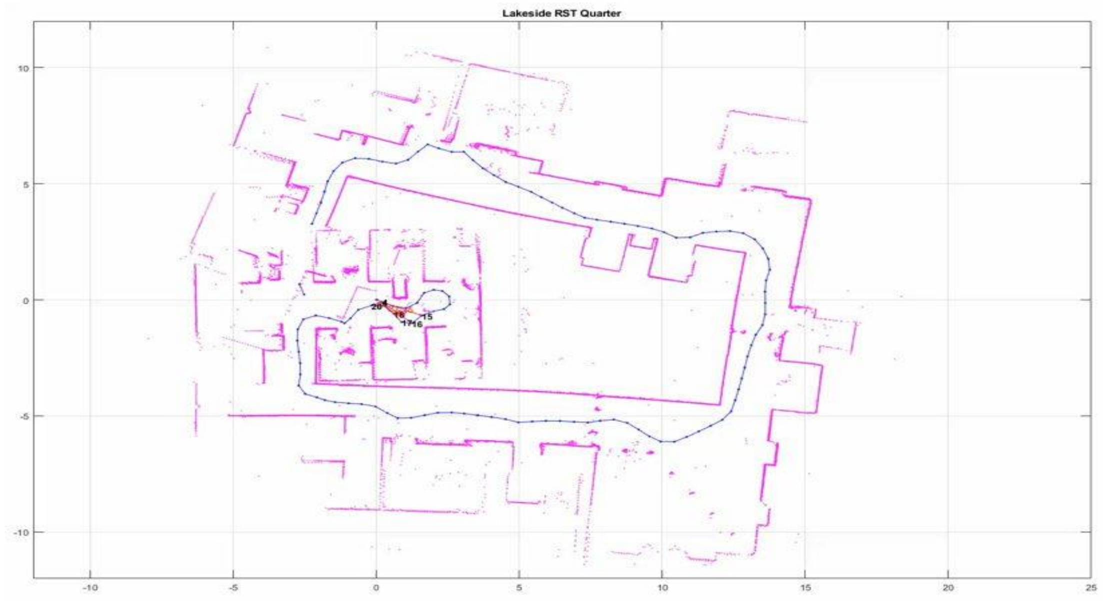


Figure 31: Example of constructing a pose graph and minimizing errors.

One countermeasure is to remember some characteristics from a previously visited place as a landmark and minimize the localization error. Pose graphs are constructed to help correct the errors. By solving error minimization as an optimization problem, more accurate map data can be generated. This kind of optimization is called bundle adjustment in visual SLAM.

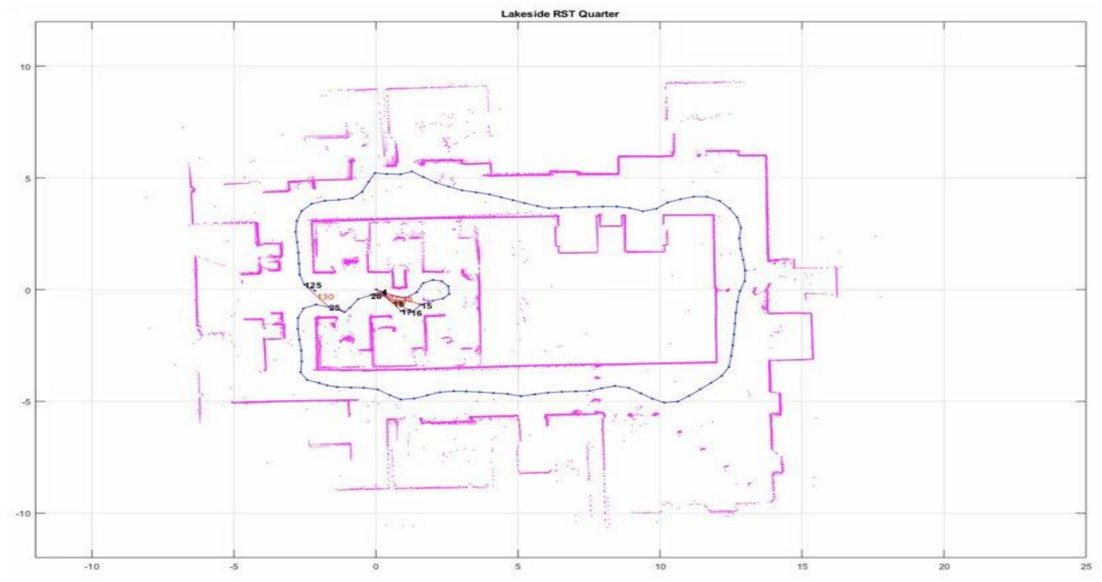


Figure 32: Example of constructing a pose graph and minimizing errors.

6.7.2 Position on the map is lost:

Image and point-cloud mapping does not consider the characteristics of a robot's movement. In some cases, this approach can generate discontinuous position estimates. For example, a calculation result showing that a robot moving at 1 m/s suddenly jumped forward by 10 meters. This kind of localization failure can be prevented either by using a recovery algorithm or by fusing the motion model with multiple sensors to make calculations based on the sensor data.

There are several methods for using a motion model with sensor fusion. A common method is using Kalman filtering for localization. Since most differential drive robots and four-wheeled vehicles generally use nonlinear motion models, extended Kalman filters and particle filters (Monte Carlo localization) are often used. More flexible Bayes filters such as unscented Kalman filters can also be used in some cases. Some commonly used sensors are inertial measurement devices such as IMU, Attitude and Heading Reference System or AHRS, Inertial Navigation System or INS, accelerometer sensors, gyro sensors, and magnetic sensors). Wheel encoders attached to the vehicle are often used for odometry. When localization fails, a countermeasure to recover is by remembering a landmark as a key-frame from a previously visited place. When searching for a landmark, a feature extraction process is applied in a way that it can scan at high speeds. Some methods based on image features include bag of features (BoF) and bag of visual words (BoVW). More recently, deep learning is used for comparison of distances from features.

6.8 SLAM limitations:

Computing cost is a problem when implementing SLAM on a vehicle hardware. Computation is usually performed on compact and low-energy embedded microprocessors that have limited processing power. To achieve accurate localization, it is essential to execute image processing and point cloud matching at high frequency. In addition, optimization calculations such as loop closure are high computation processes. The challenge is how to execute such computationally expensive processing on embedded microcomputers.

Chapter 7

System Design

7.1 System concept:

The primary modes of operation for this UGV will be automatic or self-contained, and manual or in partnership with humans. The UGV's architecture is built on four core concepts: autonomous course planning, 3D mapping using SLAM, object and weapon recognition, and search and rescue, to name a few. The design process for the UGV was broken down into three stages. In the first phase the rover was instrumented for Autonomous path planning. In the second phase the rover will operate by the implementation of Hybrid SLAM system by combining LiDAR SLAM and and visual SLAM. The third phase of the UGV consists of subsystems that were: object and weapon detection system, positioning system, sensor system, navigation system, computer vision navigation system and communication system. Naturally, there is also a need for an interface for human-machine interaction (HMI). In case the UGV is used for real world manual controlled missions, we will implement a Radio controlled system for such operations. Due to the extremely unstructured environment, it is quite conceivable for such search and rescue rovers to fall into pits or holes, become caught in small locations, or lose contact in real-world missions. We designed the rover in such a way that the cost of building it will be low. As a consequence, our system was designed with sensors that will produce the greatest results at the lowest possible cost.

7.2 System Architecture:

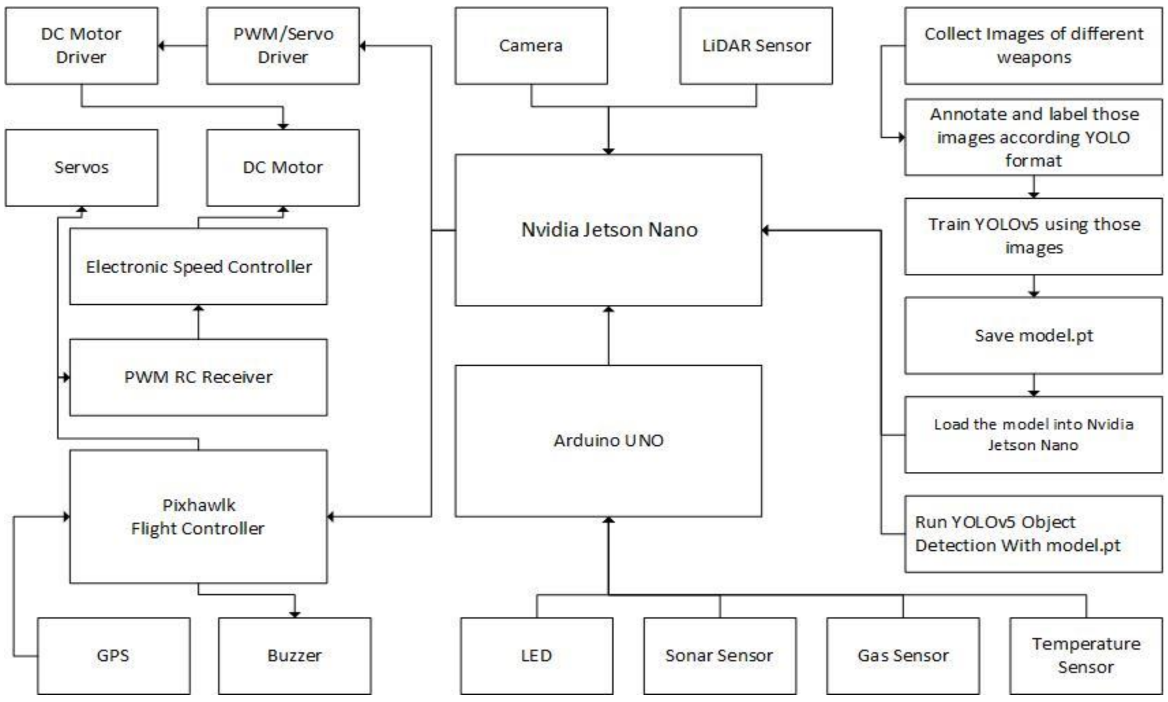

DC Motor Driver
PWM/Servo Driver
Camera
LiDAR Sensor
Collect Images of different weapons
Servos
DC Motor
Annotate and label those images according YOLO format
Nvidia Jetson Nano
Electronic Speed Controller
Train YOLOv5 using those images
Save model.pt
PWM RC Receiver
Arduino UNO
Load the model into Nvidia Jetson Nano
Pixhawlk Flight Controller
Run YOLOv5 Object Detection With model.pt
GPS
Buzzer
LED
Sonar Sensor
Gas Sensor
Temperature Sensor

Fig 35: Block diagram of the system.

7.3 Hardware Design:

Overall architecture of the UGV demonstrator was designed according to the figure 2 where the main components of the UGV demonstrator are shown:

NVidia Jetson Nano:

The NVidia Jetson Nano is a small, powerful computer packed with 128 core Maxwell GPU with CUDA and Tensorflow that lets you run multiple neural networks in parallel for applications like image classification, object detection, segmentation, and speech processing. All in an easy-to-use platform that runs in as little as 5 watts. The Nvidia Jetson Nano can also be put to use as a real-time futuristic application for countless experiments and projects in robotics, machine learning, artificial intelligence, and so much more. Users can run web servers, create simple security camera setups, design robots, develop motion detectors, and so on. With the growing need for advanced imaging and audio processing across various fields like wildlife conservation, leveraging these powerful yet compact AI supercomputers helps you unlock the maximum capabilities of your projects, prompting you to do more and better. It’s the ideal solution for prototyping your next AI-based projects.

2d LiDAR:

LiDAR is a remote sensing technology which measures the distance to an object by illuminating the target with laser and then analyzing the reflected light. This allows the sensor to calculate the correct distances between objects. For a 2D LiDAR sensor, only one laser beam is necessary. 2D sensors often use a spin movement to collect data on X and Y axes. 2D sensors are suitable for performing detection and ranging tasks. For a 3D LiDAR, the idea is the same, but several laser beams spread out on the vertical axe are shot to get data on X, Y and Z axes. 3D LiDAR applications may include terrestrial scanning and mapping. In stark contrast, high beam LiDAR involves Infrared sensors made from Indium Gallium Arsenide (InGaAs), which can cost up to tens of thousands of dollars. Though 3D sensors are more complex and expensive than 2D sensors, they can estimate the scene depth accurately.

PI cam:

The Pi camera module is a portable lightweight camera that supports Raspberry Pi. It communicates with Pi using the MIPI camera serial interface protocol. It is normally used in image processing, machine learning or in surveillance projects. It is commonly used in surveillance drones since the payload of the camera is very less. Apart from these modules Pi can also use normal USB webcams that are used along with computer. The Camera v2 is the new official camera board released by the Raspberry Pi foundation. The Raspberry Pi Camera Module v2 is a high quality 8 megapixel Sony IMX219 image sensor custom designed add-on board for Raspberry Pi, featuring a fixed focus lens.

Arduino:

Arduino Uno is a microcontroller board based on the ATmega328P (datasheet). It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz ceramic resonator (CSTCE16M0V53-R0), a USB connection, a power jack, an ICSP header and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started. You can tinker with your Uno without worrying too much about doing something wrong, worst case scenario you can replace the chip for a few dollars and start over again. "Uno" means one in Italian and was chosen to mark the release of Arduino Software (IDE) 1.0. The Uno board and version 1.0 of Arduino Software (IDE) were the reference versions of Arduino, now evolved to newer releases. The Uno board is the first in a series of USB Arduino boards, and the reference model for the Arduino platform; for an extensive list of current, past or outdated boards see the Arduino index of boards.

Flight Controller:

A flight controller (FC) is a small circuit board of varying complexity. Its function is to direct the RPM of each motor in response to input. A command from the pilot for the multi-rotor to move forward is fed into the flight controller, which determines how to manipulate the motors accordingly. The majority of flight controllers also employ sensors to supplement their calculations. These range from simple gyroscopes for orientation to barometers for automatically holding altitudes. GPS can also be used for auto-pilot or fail-safe purposes. More on that shortly. Many flight controllers allow for different flight modes, selectable using a transmitter switch. An example of a three-position setup might be a GPS lock mode, a self-leveling mode, and a manual mode. Different settings can be applied to each profile, achieving varying flight characteristics. Multiple flight modes are available: GPS lock, altitude lock, orientation mode (moving forward always happens away from take-off point, regardless of craft rotation), and a non-stabilized manual mode.

RC Transmitter & Receiver:

A Radio Transmitter is an electronic device that uses radio signals to transmit commands wirelessly via a set radio frequency over to the Radio Receiver, which is connected to an aircraft or multirotor being remotely controlled. In other words, it’s the device that translates pilot’s commands into movement of the multirotor. In some radios there is an option to connect an external transmitter module. This makes it possible to use a different frequency (for instance, 900MHz in a 2.4GHz radio) or a different receiver from another brand/protocol. A Drone Radio Transmitter transmits commands via channels. Each channel is an individual action being sent to the aircraft. Throttle, Yaw, Pitch and Roll are the four main inputs required to control the quad. Each of them uses one channel, so there is minimum of four channels required. Every switch, slider or knob on the transmitter uses one channel to send the information through to the receiver.

Radio Receivers:

A Radio Receiver is the device capable of receiving commands from the Radio Transmitter, interpreting the signal via the flight controller where those commands are converted into specific actions controlling the aircraft. Radio Receivers can have the following features:

- Telemetry (sending data back to transmitter)
- Redundancy function (two receivers connected together, if one loses connection, second one takes over)
- Easy removable antennas (more convenient with connectors if antenna is to be replaced)
- Possibility of firmware upgrades (for bug fixes) Protocols Radio communication protocols can split into two groups:
- TX Protocols between Radio Transmitter and Radio Receiver.
- RX Protocols between Radio Receiver and Flight Controller.

GPS:

GPS waypoints are much more specific: They identify an exact spot on Earth. Waypoints in the atmosphere or outer space usually include altitude. These waypoints let pilots know how tall a mountain is, for example. Users identify and mark waypoints on base maps, and can identify more than one to create a route. GPS sensors are receivers with antennas that use a satellite-based navigation system with a network of 24 satellites in orbit around the earth to provide position, velocity, and timing information. Global Positioning Systems (GPS) help us find where we are and where we’re going. They can also do the same for rovers.

Gear Motor:

Gear motors are simple DC Motors featuring gears for the shaft for obtaining the optimal performance characteristics. They are known as Center Shaft DC Geared Motors because their shaft extends through the center of their gearbox assembly. A gear motor is an all-in-one combination of a motor and gearbox. The addition of a gear head to a motor reduces the speed while increasing the torque output. The most important parameters in regards to gear motors are speed (rpm), torque (lb-in) and efficiency (%). In order to select the most suitable gear motor for application one must first compute the load, speed and torque requirements for that particular application. A motors performance and gearbox performance are combined by three specific parameters. These three parameters are speed, torque and efficiency. These performance curves are essential when selecting a gear motor for your application.

ESC:

An electronic speed controller can be designed with three essential components like a voltage regulator/ BEC (Battery Eliminator Circuit)), a Processer & the switching includes FETs. The BEC is a separation of the electronic speed control that will transmit power back to your receiver after that to servos. This also includes one secondary function like when the motor is operated through a battery then the motor gets its smallest voltage, then the BEC will keep some power for the flight control in dangerous situations so the motor doesn’t consume total the power from the battery. At present, the processor is completely enclosed within a single Si semiconductor chip. The main function of this processor is to decode the data being provided to it from the receiver within the model as well as to regulate the power toward the motor using FETs. In an ESC, this transistor plays a key role by performing all the works. It observes the complete current & voltage of the motor as well as a battery. This transistor works like a switch to control the current flow to throttle the electric motor. An ESC or electronic speed control mainly follows a speed reference signal to change the speed of a switching network of field-effect transistors. The motor speed can be changed by changing the switching frequency or the duty cycle of the transistors.

Brushed ESC:

This is a typical modern ESC for Brushed Motors. You will see it has 4 thick wires, the Red and Black are the power connectors to the battery and the Blue and Yellow wires connect to the motor. There is also a thin three wire cable that connects to your Receiver and another wire to the on off switch. ESC’s such as this are simple to connect and easy to install.

Motor drivers:

Motor drivers acts as an interface between the motors and the control circuits. Motor require high amount of current whereas the controller circuit works on low current signals. So the function of motor drivers is to take a low-current control signal and then turn it into a higher-current signal that can drive a motor. There are many Different motor drivers for Autonomous Robots. Autonomous robots are controlled basically by a Micro-controller. As mentioned above the Fire blade and Viper Motor drivers can be used to interface with Microcontrollers. Apart from these there are a few more low current motor drivers such as L293D, L298N L293D is used in various shields like Auto Shield, Xbee Shield, Starter Shield as it is best suitable for low current motors such as BO motors commonly used in small autonomous robots. L298N board is a dual motor driver which can help you interface motors which draw up to 2 Ampere of current such as BO motors and Basic Gear motors.

Battery:

Robot battery is the power source that drives all the systems on our rover and allows it to fly. Like gas in a car, once we run out of power, we are not going anywhere. When we first start driven, we need to have a basic understanding of how a robot battery works. Lithium Polymer Batteries Lithium polymer batteries are modern batteries that are commonly used in consumer electronics such as our smartphone, laptops, and tablets. The two main possible alternatives to LiPo batteries are Nickel Metal Hydride (NiMH) and Nickel Cadmium (NiCd) batteries. LiPo (lithium polymer) batteries offer significant advantages over other types of batteries. The advantages that LiPo batteries offer over NiCd and NiMH batteries are:

- LiPo batteries have higher capacities which allows them to hold more power
- LiPo batteries have higher discharge rates allowing faster power transferLiPo batteries are lighter and can be made in different shapes and sizes However, LiPo batteries also have some drawbacks. These include:
- LiPo batteries have a shorter lifespan of about 300 to 400 cycles, as compared to NiMH and NiCd batteries
- If the battery gets punctured and vents into the air, there is a possibility that this could result in a fire
- Some extra care needs to be taken when charging, discharging, or storing LiPo batteries.

Temperature sensor:

Ground platforms are powerful tools for understanding environmental conditions. Get real time data from anywhere our rover can take it, including the current temperature and relative humidity conditions. Data logging and archiving accurate sensor data for later reference can also help with overall data analysis and interpretation. Understanding the bigger environmental conditions can help you infer details about how crops and industrial processes are affected by air pollution and other parameters All LM35 family members work on the same principles but temperature measuring capacity varies and also they are available in many packages (SOIC, TO-220, TO-92,TO). LM35 can measure from -55 degrees centigrade to 150-degree centigrade. The accuracy level is very high if operated at optimal temperature and humidity levels. The conversion of the output voltage to centigrade is also easy and straight forward.

Ultra Sonic Sonar Sensor:

An ultrasonic sensor is an electronic device that measures the distance of a target object by emitting ultrasonic sound waves, and converts the reflected sound into an electrical signal. Ultrasonic waves travel faster than the speed of audible sound (i.e. the sound that humans can hear). Ultrasonic sensors have two main components: the transmitter (which emits the sound using piezoelectric crystals) and the receiver (which encounters the sound after it has travelled to and from the target). In order to calculate the distance between the sensor and the object, the sensor measures the time it takes between the emission of the sound by the transmitter to its contact with the receiver. The formula for this calculation is $D = \frac{1}{2} T \times C$ (where D is the distance, T is the time, and C is the speed of sound ~ 343 meters/second). For example, if a scientist set up an ultrasonic sensor aimed at a box and it took 0.025 seconds for the sound to bounce back, the distance between the ultrasonic sensor and the box would be:

$$\mathbf{D = 0.5 \times 0.025 \times 343}$$

or about 4.2875 meters. Ultrasonic sensors are used primarily as proximity sensors. They can be found in automobile self-parking technology and anti-collision safety systems. Ultrasonic sensors are also used in robotic obstacle detection systems, as well as manufacturing technology. In comparison to infrared (IR) sensors in proximity sensing applications, ultrasonic sensors are not as susceptible to interference of smoke, gas, and other airborne particles (though the physical components are still affected by variables such as heat). Ultrasonic sensors are also used as level sensors to detect, monitor, and regulate liquid levels in closed containers (such as vats in chemical factories). Most notably, ultrasonic technology has enabled the medical industry to produce images of internal organs, identify tumors, and ensure the health of babies in the womb.

Gas Sensor:

A gas sensor is a device that senses the atmosphere's presence or concentration of gases. The sensor creates a corresponding potential difference depending on the concentration of the gas by adjusting the resistance of the material within the sensor, which can be determined as the output voltage. The type and concentration of the gas can be calculated based on this voltage value. The type of gas that can be detected by the sensor depends on the sensing material within the sensor. Gas sensors are usually available as modules with comparators. A specific threshold value of the concentration of gas may be set for these comparators. The automated pin goes up when the concentration of the gas reaches this threshold. It is possible to use the analog pin to measure the gas concentration.

LED:

The LED is a specialized form of PN junction that uses a compound junction. The semiconductor material used for the junction must be a compound semiconductor. The commonly used semiconductor materials including silicon and germanium are simple elements and junction made from these materials do not emit light. Instead compound semiconductors including gallium arsenide, gallium phosphide and indium phosphide are compound semiconductors and junctions made from these materials do emit light. These compound semiconductors are classified by the valence bands their constituents occupy. For gallium arsenide, gallium has a valency of three and arsenic a valency of five and this is what is termed a group III-V semiconductor and there are a number of other semiconductors that fit this category. It is also possible to have semiconductors that are formed from group III-V materials. The light emitting diode emits light when it is forward biased. When a voltage is applied across the junction to make it forward biased, current flows as in the case of any PN junction. Holes from the p-type region and electrons from the n-type region enter the junction and recombine like a normal diode to enable the current to flow. When this occurs energy is released, some of which is in the form of light photons.

Buzzer:

Sensor-Buzzer is a passive buzzer. Like a magnetic speaker, it needs voltage with different frequency so that it can make sound accordingly. The pitch becomes louder when the frequency gets higher.

Chassis:

For the platform we will use The Devastator tank mobile robot platform which is constructed from high strength aluminum alloy and is extremely solid and durable. The Body shape of the chassis is designed the way that it can be construct easily and accommodate with more modules. Modular wiring – no soldering iron needed in this chassis.

Chapter 8

Reference

[1] S.T.Jawaid and S. L. Smith, "Informative path planning as a maximum traveling salesman problem with submodular rewards," Discrete Appl. Math., vol. 186, pp. 112–127, 2015.
[2] A.Krause, A. Singh, and C. Guestrin, "Near-optimal sensor placements in Gaussian processes: Theory, efficient algorithms and empirical studies," J. Mach. Learn. Res., vol. 9, pp. 235–284, 2008.
[3] N.Cao, K.H.Low, and J.M. Dolan, "Multi-robot informative path planning for active sensing of environmental phenomena: A tale of two algorithms," in Proc. 2013 Int. Conf. Auton. Agents Multi-Agent Syst., 2013, pp. 7–14.
[4] A.Singh, A.Krause, C.Guestrin, and W.J.Kaiser, "Efficient informative sensing using multiple robots," J. Artif. Intell. Res., vol. 34, no. 1, pp. 707–755, 2009.
[5] J.Yu, M.Schwager, and D.Rus, "Correlated orienteering problem and its application to informative path planning for persistent monitoring tasks," in Proc. 2014 IEEE/RSJ Int. Conf. Intell. Robots Syst., 2014, pp. 342–349.
[6] D.J.Mulla, "Twenty five years of remote sensing in precision agriculture: Key advances and remaining knowledge gaps," Biosyst. Eng., vol. 114, no. 4, pp. 358–371, 2013, special issue: Sensing Technologies for Sus- tainable Agriculture.
[7] J.S.Mitchell, "Guillotine subdivisions approximate polygonal subdivisions: A simple polynomial-time approximation scheme for geometric TSP, k-MST, and related problems," SIAM J.Comput., vol.28, no.4, pp. 1298–1309, 1999.
[8] "Clearpath robotics," 2013. [Online]. Available: http://clearpathrobotics .com. Accessed Jan. 2013.
[9] Spectrum Technologies Inc., 2013. [Online]. Available: http://www.specmeters.com/. Accessed Mar. 2013.
[10] "Mikrokopter," 2013. [Online]. Available: http://mikrokopter.de. Accessed Jan. 2013.
[11] D.Mulla, M.Beatty, and A.Sekely, "Evaluation of remote sensing and targeted soil sampling for variable rate application of nitrogen," in Proc. 5th Int. Conf. Precis. Agriculture, 2000, pp. 1–14.
[12] C.E.Rasmussen and H.Nickisch, "Gaussian processes for machine learning (GPML) toolbox," J. Mach. Learn. Res., vol. 11, pp. 3011–3015, 2010.
[13] A.Krause, "SFO: A toolbox for submodular function optimization," J. Mach. Learn. Res., vol. 11, pp. 1141–1144, 2010.
[14] D.Applegate, R.Bixby, V.Chvatal, and W.Cook, "Concorde TSP solver," 2006. [Online]. Available: http://www.tsp.gatech.edu/concorde

[15] “Tetracam,” 2014. [Online]. Available: www.tetracam.com. Accessed Feb. 2014.

[16] G. Zhang, C. A. Duncan, J. Kanno, and R. R. Selmic “Unmanned Ground Vehicle Navigation in Coordinate-Free and Localization-Free Wireless Sensor and Actuator Networks,” in Journal of Intelligent & Robotic Systems, vol. 74, Issue 3-4, 2014, pp. 869-891.

[17] P.Velaskar, A.Vargas-Clara, O.Jameel, and S.Redkar “Guided Navigation Control of an Unmanned Ground Vehicle using Global Positioning Systems and Inertial Navigation Systems,” in International Journal of Electrical and Computer Engineering (IJECE), vol. 4, Issue 3, 2014, pp. 329-342, 2014.

[18] army-technology, “Oshkosh TerraMax Unmanned Ground Vehicle Technology,” [Online]:http://www.armytechnology.com/projects/oshkosh-terramax-unmanned-ground vehicle/.

[19] Segfault team–Dalhousie University, “Robot Design Report,” [online]: http://www.igvc.org/design/2013/Dalhousie%20University.pdf

[20] R. R. Murphy, “Introduction to AI robotics,” The MIT Press, 2000.

[21] E.W.Dijkstra, “A note on two problems in connection with graphs,” in Numerische Mathematik pp. 269–271, 1959.

[22] C.E.Leiserson and T.B.Schardl, “A Work-Efficient Parallel BreadthFirst Search Algorithm,” in ACM Symposium on Parallelism in Algorithms and Architectures, 2010.

[23] P. E. Hart, N.J.Nilsson, and B.Raphael, “A Formal Basis for the Heuristic Determination of Minimum Cost Paths,” in IEEE Transactions on Systems Science and Cybernetics SSC4, vol. 4, Issue 2, 1968, pp. 100–107.

[24] M.Likhachev, D.Ferguson, and G.Gordon, “Anytime Dynamic A*: An Anytime, Replanning Algorithm.,” ICAPS, 2005.

[25] Rutuja. R. Bachhav, Priyanka. S. Bhavsar, Devyani. K. Dambhare, “Spy Robot For Video Surveillance And Metal Detection” In International Journal For Research In Applied Science & Engineering Technology (Ijraset); Issn: 2321-9653; Volume 6 Issue V, May 2018.

[26] Neha Joisher, Sonaljain, Tejas Malviya, Vaishali Gaikwad (Mohite), “Surveillance System Using Robotic Car” In International Conference On Control, Automation And Systems, Ieee 2018.

[27] Fatma Salih, Mysoon S.A. Omer, “Raspberry Pi As A Video Server” In The International Conference On Computer, Control, Electrical, And Electronics Engineering (Iccceee) 2018.

[28] Sweeta Deshmukh, Priyadarshini, Mamta, Madhura Deshmukh, Dr.Md. Bakhar, “Iot Based Surveillance Robot” In International Journal Of Innovative Research In Computer And Communication Engineering; Vol.5, Special Issue 4, June 2017

[29] Ghanem Osman Elhaj Abdalla, T. Veeramanikandasamy, “Implementation Of Spy Robot For A Surveillance System Using Internet Protocol Of Raspberry Pi” In 2nd Ieee International Conf On Recent Trends In Electronics Information & Communication Technology, May 19-20, 2017.

[30] Souvik Saha, Arko Singh, Palash Bera, Murshed Nawaz Kamal, Soumya Dutta, Uttam Gorian, Sayak Pramanik, Angshuman Khan, Surajit Sur,
“Gps Based Smart Spy Surveillance Robotic System Using Raspberry Pi For Security Application And Remote Sensing” In Ieee , 978-1-53863371-7/17,2017.

[31] Zeeshan Shaikh, Priyanka Gaikwad, Nikhil Kare, Swapnali Kapade, M.V. Korade , “An Implementation On – Surveillance Robot Using Raspberry-Pi Technology” In International Research Journal Of Engineering And Technology (Irjet); Volume: 04 Issue: 04; April 2017.

[32] Priya B. Patel, Viraj M. Choksi, Swapna Jadhav, M.B. Potdar, “Smart Motion Detection System Using Raspberry Pi” In International Journal Of
Applied Information Systems (Ijais);Foundation Of Computer Science Fcs, New York, Usa; Volume 10 – No.5, February 2016.

[33] C.J.Ostafew, A. P. Schoellig, T. D. Barfoot, and J. Collier, ‘‘Learningbased nonlinear model predictive control to improve vision-based mobile robot path tracking,’’ J. Field Robot., vol. 33, no. 1, pp. 133–152, 2015.

[34] Y. LeCun, Y. Bengio, and G. Hinton, ‘‘Deep learning,’’ Nature, vol. 521, no. 7553, p. 436, 2015.

[35] R.S.Sutton and A.G.Barto, Reinforcement Learning: An Introduction, vol. 1, no. 1. Cambridge, MA, USA: MIT Press, 1998.

[36] C.Chen, A.Seff, A.Kornhauser, and J.Xiao, ‘‘DeepDriving: Learning affordance for direct perception in autonomous driving,’’ in Proc. IEEE Int. Conf. Comput. Vis., Dec. 2015, pp. 2722–2730.

[37] T. Kollar and N. Roy, ‘‘Trajectory optimization using reinforcement learning for map exploration,’’ Int. J. Robot. Res., vol. 27, no. 2, pp. 175–196, 2008.

[38] D. Silver, G. Lever, N. Heess, T. Degris, D. Wierstra, and M. Riedmiller, ‘‘Deterministic policy gradient algorithms,’’ in Proc. 31st Int. Conf. Mach. Learn. (ICML), 2014, pp. 387–395.

[39] S. Paul and L. Vig, ‘‘Deterministic policy gradient based robotic path planning with continuous action spaces,’’ in Proc. IEEE Int. Conf. Comput. Vis. Workshops (ICCVW), Oct. 2017, pp. 725–733.
[40] S. Gu, E. Holly, T. Lillicrap, and S. Levine, ‘‘Deep reinforcement learning for robotic manipulation with asynchronous off-policy updates,’’ in Proc. IEEE Int. Conf. Robot. Automat. (ICRA), May/Jun. 2017, pp. 3389–3396.
[41] A. Nair, B. McGrew, M. Andrychowicz, W. Zaremba, and P. Abbeel. (2017). ‘‘Overcoming exploration in reinforcement learning with demonstrations.’’ [Online]. Available: https://arxiv.org/abs/1709.10089
[42] L. Tai, G. Paolo, and M. Liu. (2017). ‘‘Virtual-to-real deep reinforcement learning: Continuous control of mobile robots for mapless navigation.’’ [Online]. Available: https://arxiv.org/abs/1703.00420
[43] P. Mirowski et al. (2016). ‘‘Learning to navigate in complex environments.’’ [Online]. Available: https://arxiv.org/abs/1611.03673
[44] Y. Duan, X. Chen, R. Houthooft, J. Schulman, and P. Abbeel, ‘‘Benchmarking deep reinforcement learning for continuous control,’’ in Proc. Int.
Conf. Mach. Learn., 2016, pp. 1329–1338.
[45] Wang, Qi, et al, "Automated crop yield estimation for apple orchards", Experimental robotics. Springer, p. 745-758, 2013.
[46] Wang, Zhenglin, Kerry Walsh, and Brijesh Verma, "On-tree mango fruit size estimation using RGB-D images", Sensors, 17.12: 2738, 2017.
[47] Stajnko, Denis, Miran Lakota, and Marko Hoˇcevar, "Estimation of number and diameter of apple fruits in an orchard during the growing season by thermal imaging", Computers and Electronics in Agriculture, 42.1: 31-42, 2004.
[48] Regunathan, Murali, andWon Suk Lee, "Citrus fruit identification and size determination using machine vision and ultrasonic sensors", ASAE Annual Meeting. American Society ofAgricultural and Biological Engineers, p. 1, 2005.
[49] Kamilaris, Andreas, and Francesc X. Prenafeta-Boldú, "Deep learning in agriculture: A survey", Computers and electronics in agriculture, 147: 7090, 2017.
[50] Tang, J.L., Wang, D., Zhang, Z.G., He, L.J., Xin, J., Xu, Y., 2017. Weed identification based on K-means feature learning combined with convolutional neural network. Comput. Electron. Agric. 135, 63–70.

[51] Zhang, Y.D., Dong, Z., Chen, X., Jia, W., Du, S., Muhammad, K., Wang, S.H., 2017. Image based fruit category classification by 13-layer deep convolutional neural network and data augmentation. Multimedia Tools Appl. 1–20.

[52] Dias, P.A., Tabb, A., Medeiros, H., 2018. Apple flower detection using deep convolutional networks. Comput. Ind. 99, 17–28.

[53] Bargoti, S., Underwood, J., 2016. Deep fruit detection in orchards. Aust. Centre Field Robotics 1–8.

[54] Yamamoto, K., Guo, W., Yoshioka, Y., Ninomiya, S., 2014. On plant detection of intact tomato fruits using image analysis and machine learning
methods. Sensors 14, 12191–12206.

[55] ASTM Committee F45 Driverless Automatic Guided Industrial Vehicles, www.astm.org/F45, accessed January 26, 2018.

[56] Josh Bond, "AGVs: Predictably http://www.mmh.com/article/agvs_predictably_flexible, 2018, Accessed January 26, 2018. January Flexible", 19,

[57] International Organization of Standards (ISO) 8373:2012 Robots and robotic devices – Vocabulary, 2012.

[58] Merriam-Webster, https://www.merriam-webster.com/dictionary/, accessed 1/24/2018.

[59] Google, www.google.com, accessed 1/24/2018. [6] Wikipedia, www.wiki.com, accessed 1/24/2018. [7] L.G. Shattuck, "Transitioning to Autonomy: A human systems integration perspective", Presentation at Transitioning to Autonomy: Changes in the role of humans in air transportation, March 11, 2015. Available at human-actors.arc.nasa.gov/workshop/autonomy/download/presentations/Shaddo ck%20.pdf (Accessed November 3, 2017.)

[60] Hui-Min Huang, Kerry Pavek, James Albus, Elena Messina, "Autonomy Levels for Unmanned Systems (ALFUS) Framework: An Update", 2005 SPIE Defense and Security Symposium, Orlando, Florida, 2005.

[62] Ryan W. Proud, Jeremy J. Hart, and Richard B. Mrozinski, "Methods for Determining the Level of Autonomy to Design into a Human Spaceflight Vehicle: A Function Specific Approach, https://ntrs.nasa.gov/archive/nasa/casi.ntrs.nasa.gov/20100017272.pdf, accessed November 2, 2017.

[63] Lieutenant Colonel Jeffrey N. Rule, "A Symbiotic Relationship: The OODA Loop, Intuition, and Strategic Thought", Strategy Research Project, US Army War College, http://www.dtic.mil/dtic/tr/fulltext/u2/a590672.pdf, accessed January 26, 2018.Bogoslavskyi,

I., Stachniss, C., 2016. Fast range image-based segmentation of sparse 3D laser scans for online operation. International Conference on Intelligent Robots and Systems (IROS) 163 169.
[64] Antoni Burguera, Gabriel Oliver, and Juan D Tardos. Robust scan matching localization using ultrasonic range finders. In Intelligent Robots and Systems, 2005.(IROS 2005). 2005 IEEE/RSJ International Conference on, pages 1367–1372. IEEE, 2005.
[65] Antoni Burguera, Yolanda Gonz´alez, and Gabriel Oliver. Sonar sensor models and their application to mobile robot localization. Sensors,9 (12):10217–10243, 2009.
[66] Oscar De Silva, George KI Mann, and Raymond G Gosine. Development of a relative localization scheme for ground-aerial multi-robot systems. In Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on, pages 870–875. IEEE, 2012
[67] Ethan Eade and Tom Drummond. ing, 27(5):588 – 596, 2009. Edge landmarks in monocular {SLAM}. ISSN 0262-8856. Image and Vision Computdoi: http://dx.doi.org/10.1016/j.imavis.2008.04.012. URL http://www.sciencedirect.com/science/article/pii/S0262885608000978. (BMVC 2006).
The 17th British Machine Vision Conference
[68] Felix Endres, J¨urgen Hess, J¨urgen Sturm, Daniel Cremers, and Wolfram Burgard. 3-d mapping with an rgb-d camera. 2014. 6. Arturo Gil, Oscar´ Reinoso, M´onica Ballesta, and Miguel Juli´a. Multi-robot visual slam using a rao-blackwellized particle filter. Robotics and Autonomous Systems, 58(1):68–80, 2010.
[69] Arturo Gil, scar Reinoso, Mnica Ballesta, and Miguel Juli. Multi-robot visual {SLAM} using a rao-blackwellized particle filter. Robotics and Autonomous Systems, 58(1):68 – 80, 2010. ISSN 0921-8890. doi: http://dx.doi.org/10.1016/j.robot.2009.07.026. URL http://www.sciencedirect.com/science/article/pii/S0921889009001158.
[70] H Gonzalez-Jorge, B Riveiro, E Vazquez-Fernandez, J Mart´ınez-S´anchez, and P Arias. Metrological evaluation of microsoft kinect and asus xtion sensors. Measurement, 46(6):1800–1806, 2013.
[71] Jingwei Guan and MQ-H Meng. Study on distance measurement for nao humanoid robot. In Robotics and Biomimetics (ROBIO), 2012 IEEE International Conference on, pages 283 286. IEEE, 2012.
[72] Jungong Han, Ling Shao, Dong Xu, and Jamie Shotton. Enhanced computer vision with microsoft kinect sensor: A review. 2013. 11. Il-Kyun Jung and Simon Lacroix. High resolution terrain mapping using low attitude aerial stereo imagery. In Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on, pages 946–951. IEEE, 2003.

[73] Michael Kaess and Frank Dellaert. Probabilistic structure matching for visual slam with a multi-camera rig. Computer Vision and Image Understanding, 114(2):286–296, 2010.

[74] Kourosh Khoshelham and Sander Oude Elberink. Accuracy and resolution of kinect depth data for indoor mapping applications. Sensors,12 (2):1437–1454, 2012

[75] T. E. Bishop and P. Favaro. The light field camera: Extended depth of field, aliasing, and superresolution. IEEE Transactions on Pattern Analysis and Machine Intelligence, 34(5):972–986, 2012.

[76] J. L. Blanco, J. A. Fern´andez-Madrigal, and J. Gonzalez. A Novel Measure of Uncertainty for Mobile Robot SLAM with RaoBlackwellized Particle Filters. The International Journal ofRobotics Research (IJRR), 27(1):73–89, 2008.

[78] J. Bloomenthal. Introduction to Implicit Surfaces. Morgan Kaufmann, 1997.

[79] A. Bodis-Szomoru, H. Riemenschneider, and L. Van-Gool. Efficient edge-aware surface mesh reconstruction for urban scenes. Journal of Computer Vision and Image Understanding, 66:91–106, 2015.

[80] M. Bosse, P. Newman, J. J. Leonard, and S. Teller. Simultaneous Localization and Map Building in Large-Scale Cyclic Environments Using the Atlas Framework. The International Journal of Robotics Research (IJRR), 23(12):1113–1139, 2004.

[81] M. Bosse and R. Zlot. Continuous 3D Scan-Matching with a Spinning 2D Laser. In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), pages 4312–4319. IEEE, 2009.

[82] M. Bosse and R. Zlot. Keypoint design and evaluation for place recognition in 2D LIDAR maps. Robotics and Autonomous Systems (RAS), 57(12):1211–1224, 2009.

[83] M. Bosse, R. Zlot, and P. Flick. Zebedee: Design of a SpringMounted 3D Range Sensor with Application to Mobile Mapping. IEEE Transactions on Robotics (TRO), 28(5):1104–1119, 2012.

[84] F. Bourgault, A. A. Makarenko, S. B. Williams, B. Grocholsky, and H. F. Durrant-Whyte. Information Based Adaptive Robotic Exploration. In Proceedings ofthe IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pages 540–545. IEEE, 2002.

[85] C. Brand, M. J. Schuster, H. Hirschmuller, and M. Suppa. Stereo-Vision Based Obstacle Mapping for Indoor/Outdoor SLAM. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pages 2153–0858. IEEE, 2014.

[86] W. Burgard, M. Moors, C. Stachniss, and F. E. Schneider. Coordinated Multi-Robot Exploration. IEEE Transactions on Robotics (TRO),
21(3):376–386, 2005
[87] Li J, Wu Z, Zhang J, et al. Research of AGV positioning based on the two-dimensional Code Recognition Method[C]// International Conference on Logistics, Informatics and Service Sciences. IEEE, 2015