A Major Project Report

On

**Tomato Crop Monitoring using Rover**

Submitted in partial fulfillment of the

Requirements for the award of degree of

**Bachelor of Technology**

In

**Computer Science and Engineering**

by

**LAKKAM RAMYA SREE**

**[17H61A05M5]**

**Under the Guidance of**

**G. BALRAM**

**Assistant Professor**



**Department of Computer Science and Engineering**

**ANURAG GROUP OF INSTITUTIONS**

**(Formerly CVSR College of Engineering)**

**(An Autonomous Institution, Approved by AICTE and NBA Accredited)**

**Venkatapur (V), Ghatkesar (M), Medchal(D)., T.S-500088**

**(2017-2021)**

****

**CERTIFICATE**

This is to certify that the project entitled **“ TOMATO CROP MONITORING USING ROVER ”**  being submitted by **LAKKAM RAMYA SREE** bearing the Hall Ticket number **17H61A05M5** in partial fulfillment of the requirements for the award of the degree of the **Bachelor of Technology** in **Computer Science and Engineering** to **Anurag Group of Institutions** **(Formerly** **CVSR College of Engineering)** is a record of bonafide work carried out by her under my guidance and supervision from April 2021 to July 2021.

The results presented in this project have been verified and found to be satisfactory. The results embodied in this project report have not been submitted to any other University for the award of any other degree or diploma.

**Internal Guide External Examiner**

**Mr. G. BALRAM**

**Assistant Professor, Dept. of CSE**

**Dr. G.Vishnu Murthy,**

**Professor and Head, Dept. of CSE**

**ACKNOWLEDGEMENT**

It is my privilege and pleasure to express profound sense of respect, gratitude and indebtedness to my guide **Mr. G. BALRAM, Assistant Professor**, Dept. of Computer Science and Engineering, Anurag Group of Institutions (Formerly CVSR College of Engineering), for his indefatigable inspiration, guidance, cogent discussion, constructive criticisms and encouragement throughout this dissertation work.

I express my sincere gratitude to **Dr. G. Vishnu Murthy, Professor & Head**, Department of Computer Science and Engineering, Anurag Group of Institutions (Formerly CVSR College of Engineering), for his suggestions, motivations and co-operation for the successful completion of the work.

I extend my sincere thanks to **Dr. V. Vijaya Kumar**, **Dean,** Research and Development, Anurag Group of Institutions, for his encouragement and constant help.

I extend my sincere thanks to **Dr. K.S. Rao, Director**, Anurag Group of Institutions for his encouragement.

**LAKKAM RAMYA SREE** (17H61A05M5)

**DECLARATION**

I hereby declare that the project work entitled “**TOMATO CROP MONITORING ROVER** ” submitted to the **Anurag Group of Institutions(Formerly CVSR College of Engineering)** in partial fulfillment of the requirements for the award of the degree of **Bachelor of Technology (B. Tech)** in Computer Science and Engineering is a record of an original work done by us under the guidance of **Mr.G. BALRAM, Assistant Professor** and this project work have not been submitted to any other university for the award of any other degree or diploma.

**LAKKAM RAMYA SREE** (17H61A05M5)

**DATE:**

**ABSTRACT**

Agriculture is that the one among the previous profession, particularly in India. today several robots and devices square measure introduced to stay the plants healthy condition and to attain full life cycle with none chemical effects to human and atmosphere. During this work automaton is meant as semiautonomous to attain associate agriculture task with additional economical and precise by victimization current trends. The main objective behind this is to develop a rover to perform the Crop Monitoring in domestic areas. Nowadays robot/rover plays a vital role in our day to day life activities thus reducing human labor and human error. Rovers can be manually controlled or can be automatic based on the requirement. The purpose of this rover is to roam around the fields and to provide video information from the given environment and to capture the images of crops and send them to a cloud platform. In the cloud using the collected pictures we apply certain inbuilt algorithms on the pictures to determine the condition of the crops. In this project, one can control the rover with the help of mobile or laptop through Internet of Things (IoT) and also can get the live streaming with the help of wireless camera from the rover. This rover also uses various sensors that collects data and sends it to the microcontroller which controls the rover behavior.

**Keywords:** Internet Of Things[IOT],ML, Rovers, Microcontroller,tensorflow.

|  |  |  |  |
| --- | --- | --- | --- |
| **TABLE OF CONTENTS** | | | |
| **S.NO** | **CONTENTS** | | **PAGE NO** |
| 1. | Introduction | | 1 |
|  | 1.1. Motivation | | 1 |
|  | 1.2. Problem Definition | | 1 |
|  | 1.3. Objective of the Project | | 2 |
| 2. | Literature Survey | | 3 |
| 3. |  | Analysis | 5 |
|  | 3.1. Existing System | | 5 |
|  | 3.2. Proposed System | | 5 |
|  | 3.3. Hardware Requirement Specification | | 6 |
|  | 3.4. Software Requirement Specification | | 8 |
|  | 3.4.2 Scope | | 8 |
|  | 3.4.3 Overall Description | | 8 |
| 4. |  | Design | 10 |
|  | 4.1. UML diagrams | | 10 |
| 5. | Implementation | | 14 |
|  | 5.1. Modules | | 14 |
|  | 5.2. Module description | | 15 |
|  | 5.3. Introduction of technologies used | | 15 |
|  | 5.4. Sample Code | | 28 |
| 6. | Test cases | | 37 |
| 7. | Screenshots | | 39 |
| 8. | Conclusion | | 43 |
| 9. | Future Enhancement | | 44 |
|  | Bibliography | | 45 |

1. **INTRODUCTION**

Robotics is a promising technology that contributes to almost every sector of the global economy, from medical to space study. Nevertheless, one sector that consistently lags behind is the agriculture. This is slightly mystifying because many farmers were used to the tools, heavy machinery and conventional agricultural techniques. The usage of robotics related and automation technologies provides significant values to both farmers and agricultural field. These tools are used for common applications like plant classification, etc. Automated agricultural tasks bring many benefits to the field which avoids unexpected or dangerous effects of chemical exposure which also helps to improve overall efficiency and productivity. Automation enables many advantages to farmers or land owners which makes the task held more accurate, uniform and less costly.

By interfacing Wi-Fi module with Arduino, we can get unlimited range of operation. Rover can be operated in both manual and automatic modes. The communication with the robot occurs in a more secured manner. By using Arduino microcontroller, the cost and complexity can be reduced.

* 1. **Problem Definition**

Agriculture is an important sector of Indian economy. So in Agriculture the growth of crops is the main important thing. As several crops are grown at a time it sometimes becomes difficult to monitor the growth of all the crops. So here crop monitoring rover helps to monitor the tomato crops from a single place thereby reducing the efforts of farmers. This rovers checks the physical conditions of crops.

* 1. **Objective of the Project**

Agriculture is that the one among the previous profession, particularly in India, one sector that consistently lags behind is the agriculture.To Improve the Agriculture process in an effective way we use automated rovers which monitor the crops throughout the growing cycle and so this rovers reduces the effort of farmers to a certain extent. It enables many advantages to the farmers.

The objective of this rover is to roam around the fields and to provide video information from the given environment and to capture the images of crops and store them in a local folder. Further we apply certain algorithms on the pictures to determine the condition of the crops. In this project, one can control the rover with the help of mobile or laptop through Internet of Things (IoT)

1. **LITERATURE SURVEY**

**1Manjunath P M, 2Gurucharan, 3Shwetha, 4Prof. Melwin D Souza 1Student, 2Student, 3Student,4Senior Assistant Professor 1Computer Science and Engineering 1Moodlakatte Institute of Technology, Kundapur, India**

Nearly three quarters of Indian families depend on agricultural income. Almost all crops suffer from plant diseases and insects like Tungro virus, moths and butterflies. Farmers find it difficult to identify actual disease. According to Indian crop survey, on an average scale of 80 percent of crops get damaged due to disease and insects. If the problems are known before then diseases can be prevented. There is a scope for improvement in these fields using IoT Technology. So, we are designing an IoT based robot that will monitor the crop and also the environment around the crop. This system uses machine learning technique to identify the problem and takes measures to prevent diseases and insects that harm the crops. Different sensors are used to study the environment, and a camera to detect the plant type and disease. A number of prototype guidance systems have been developed but have not yet proceeded to commercialization. Our crop monitoring system is efficient and affordable which will help the Indian farmers.

**1K.Komal Devi, 2Dr.Josephine Premkumar, 3K.Kavitha, 4 P.Anitha, 5M. Sathish Kumar, 6R.Mahaveerakannan 1,3,4,5,6Assistant Professor, Department of Information Technology, Hindusthan College of Engineering and Technology, Coimbatore, India. 2 Professor, Department of Computer Science and Engineering, Cambridge Institute of Technology, Bangalore,India.**

Farming can be made more productive and accurate with the help of technologies like IOT. IOT can be used in different domains of farming or agriculture. The area of implementation of IOT is wide and it can be implemented in every field. But still it is lacking in the field of agriculture where the technology has not been accepted due to economic conditions of the farmer. This paper is about brief survey on development of IOT in different areas of Agriculture field especially in crop monitoring. In this paper surveyed various research papers on crop monitoring in the field of agriculture. Typical agriculture methods used by farmers and their difficulties they face. This survey provides an overview of some of the key IOT challenges and its stack layers also provide a summary of related research works on crop monitoring.

|  |
| --- |
| 1. **Analysis**   **3.1. Existing System**  Already some existing systems use rovers that have limited range of communication as they are based on Zigbee and Bluetooth whose range is less when compared with the wifi. Also some existing projects use short range wireless camera which makes it difficult for live streaming in long range. Some of them can only be controlled in a manual mode which needs human supervision throughout the whole surveillance process which again creates problems like turning on and turning off the camera always when required. A rover which performs image processing using the camera on an Android smartphone has also been implemented. However, this method is limited by the processing power of the phone, a problem that we have addressed by remotely performing all imaging processing operations on a different computer, after transmitting the camera’s feed. Our project is rather unique in the sense that it is a low-cost solution that offers the ability to remotely control a robot with an unlimited range (by using the internet), while also offering video feedback. There is also no constraint on any extra processing since everything is done remotely.  **3.2. Proposed System**  The rover is specially designed for Tomato Crop Monitoring. The control mechanism is provided along with video transmission facility. The video transmission is practically achieved through highspeed image transmission. Initially, the rover will be equipped with an ESP32 camera which will capture the images of crop in front of it with the help of a telegram bot which is connected to the ESP32.The images are then stored into a local folder. Mi cam is also used in this for live monitoring and to know in which direction the rover is moving.  Later we develop crop prediction algorithm in jupyter notebook with the help of tensorflow libraries and other inbuilt modules. we train the captured images using this algorithm and based on the training efficiency the condition of the crop is predicted.    **3.3. Hardware Requirement Specification**   1. **Esp32 camera Module**   It is a type of Node MCU module. It consists of inbuilt wifi and camera module. This module runs on a 3.3v to 5v. It is connected to any device using the wifi module and with it we can communicate with devices.It is a series of low-cost, low-power [system on a chip](https://en.wikipedia.org/wiki/System_on_a_chip) [microcontrollers](https://en.wikipedia.org/wiki/Microcontroller) with integrated [Wi-Fi](https://en.wikipedia.org/wiki/Wi-Fi) and dual-mode [Bluetooth](https://en.wikipedia.org/wiki/Bluetooth). The ESP32 series employs a [Tensilica](https://en.wikipedia.org/wiki/Tensilica) Xtensa LX6 microprocessor in both dual-core and [single-core](https://en.wikipedia.org/wiki/Single-core) variations and includes built-in antenna switches, RF [balun](https://en.wikipedia.org/wiki/Balun), power amplifier, low-noise receive amplifier, filters, and power-management modules. ESP32 is created and developed by [Espressif Systems](https://en.wikipedia.org/w/index.php?title=Espressif_Systems&action=edit&redlink=1), a Shanghai-based Chinese company, and is manufactured by [TSMC](https://en.wikipedia.org/wiki/TSMC) using their 40 nm process.[[2]](https://en.wikipedia.org/wiki/ESP32#cite_note-EspressifESP32ProductPage-2) It is a successor to the [ESP8266](https://en.wikipedia.org/wiki/ESP8266) microcontroller.   1. **Arduino Uno**   Arduino Uno is a microcontroller board based on the ATmega328P . 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. Esp32 is programmed with the help of Arduino.   1. **Motor Driver IC L298D**   The L298N is an integrated monolithic circuit in a 15- lead Multiwatt and PowerSO20 packages. It is a high voltage , high current dual full-bridge driver de-signed to accept standard TTL logic level sand drive inductive loads such as relays, solenoids, DC and stepping motors. Two enable inputs are provided to enable or disable the device independently of the in-put signals.The emitters of the lower transistors of each bridge are connected together rand the corresponding external terminal can be used for the connection of an external sensing resistor. An additional Supply input is provided so that the logic works at a lower voltage.   1. **Two DC motor**   A **DC motor** is any of a class of rotary [electrical motors](https://en.wikipedia.org/wiki/Electrical_motor) that converts direct current electrical energy into mechanical energy. The most common types rely on the forces produced by magnetic fields. Nearly all types of DC motors have some internal mechanism, either electromechanical or electronic, to periodically change the direction of current in part of the motor.  DC motors were the first form of motor widely used, as they could be powered from existing direct-current lighting power distribution systems. A DC motor's speed can be controlled over a wide range, using either a variable supply voltage or by changing the strength of current in its field windings. Small DC motors are used in tools, toys, and appliances. The [universal motor](https://en.wikipedia.org/wiki/Universal_motor) can operate on direct current but is a lightweight [brushed](https://en.wikipedia.org/wiki/Brush_(electric)) motor used for portable power tools and appliances. Larger DC motors are currently used in propulsion of electric vehicles, elevator and hoists, and in drives for steel rolling mills. The advent of [power electronics](https://en.wikipedia.org/wiki/Power_electronics) has made replacement of DC motors with [AC motors](https://en.wikipedia.org/wiki/AC_motors) possible in many applications.   1. **Node MCU**   NodeMCU is an open source firmware for which open source prototyping board designs are available. The name "NodeMCU" combines "node" and "MCU" (micro-controller unit).The term "NodeMCU" strictly speaking refers to the firmware rather than the associated development kits.Both the firmware and prototyping board designs are open source.The firmware uses the Lua scripting language. The firmware is based on the eLua project, and built on the Espressif Non-OS SDK for ESP8266. It uses many open source projects, such as lua-cjson and SPIFFS. Due to resource constraints, users need to select the modules relevant for their project and build a firmware tailored to their needs. Support for the 32-bit ESP32 has also been implemented.  The prototyping hardware typically used is a circuit board functioning as a dual in-line package (DIP) which integrates a USB controller with a smaller surface-mounted board containing the MCU and antenna. The choice of the DIP format allows for easy prototyping on breadboards. The design was initially based on the ESP-12 module of the ESP8266, which is a Wi-Fi SoC integrated with a Tensilica Xtensa LX106 core, widely used in IoT applications   1. **Batteries**   A **battery** is a device consisting of one or more [electrochemical cells](https://en.wikipedia.org/wiki/Electrochemical_cell) with external connections[[1]](https://en.wikipedia.org/wiki/Electric_battery#cite_note-1) for powering [electrical](https://en.wikipedia.org/wiki/Electricity) devices such as [flashlights](https://en.wikipedia.org/wiki/Flashlight), [mobile phones](https://en.wikipedia.org/wiki/Mobile_phone), and [electric cars](https://en.wikipedia.org/wiki/Electric_car). When a battery is supplying [electric power](https://en.wikipedia.org/wiki/Electric_power), its positive terminal is the [cathode](https://en.wikipedia.org/wiki/Cathode) and its negative terminal is the [anode](https://en.wikipedia.org/wiki/Anode).[[2]](https://en.wikipedia.org/wiki/Electric_battery#cite_note-Pauling1988-2) The terminal marked negative is the source of electrons that will flow through an external electric circuit to the positive terminal. When a battery is connected to an external electric load, a [redox](https://en.wikipedia.org/wiki/Redox) reaction converts high-energy reactants to lower-energy products, and the [free-energy](https://en.wikipedia.org/wiki/Gibbs_free_energy) difference is delivered to the external circuit as electrical energy. Historically the term "battery" specifically referred to a device composed of multiple cells, however the usage has evolved to include devices composed of a single cell.  [Primary](https://en.wikipedia.org/wiki/Primary_battery) (single-use or "[disposable](https://en.wikipedia.org/wiki/Disposable_product)") batteries are used once and discarded, as the [electrode](https://en.wikipedia.org/wiki/Electrode) materials are irreversibly changed during discharge; a common example is the [alkaline battery](https://en.wikipedia.org/wiki/Alkaline_battery) used for flashlights and a multitude of portable electronic devices. [Secondary (rechargeable) batteries](https://en.wikipedia.org/wiki/Rechargeable_battery) can be discharged and recharged multiple times using an applied electric current; the original composition of the electrodes can be restored by reverse current. Examples include the [lead-acid batteries](https://en.wikipedia.org/wiki/Lead-acid_batteries) used in vehicles and [lithium-ion](https://en.wikipedia.org/wiki/Lithium-ion) batteries used for portable electronics such as [laptops](https://en.wikipedia.org/wiki/Laptop) and mobile phones.  **3.4. Software Requirement Specification**   1. Arduino Ide 2. Blynk 3. Jupyter Notebook 4. Anaconda   **3.4.2 Scope of Project**  The Future Implications are very vast of this type of technology. This technology can also be used with the Home Automation technologies. It can also be made more advanced such that it can be used to monitor any type of crop. Agriculture is an important sector of Indian economy. So in Agriculture the growth of crops is the main important thing. As several crops are grown at a time it sometimes becomes difficult to monitor the growth of all the crops. So here crop monitoring rover helps to monitor the tomato crops from a single place thereby reducing the efforts of farmers. This rovers checks the physical conditions of crops.  **3.4.3 Overall Description**  A rover is a machine that is basically place or mounted on a movable platform and can be with the help of certain instructions. In today’s world a lot of fields use mobile rovers. Many of the complex rovers that we now see have originated from the simpler mobile rovers. The combination of mobile devices and rovers are leading to new ideas in lots of fields.  This rover is specially designed for monitoring purpose of tomato crop. The control mechanism is provided along with video transmission facility. The video transmission is practically achieved through highspeed image transmission. Initially, the rover will be equipped with an ESP32 camera which will capture the images in front of it using a telegram bot connected to ESP32. Also a Mi camera is used for live monitoring and also to know in which direction the rover is moving. The captured images are further trained using algorithm to determine crop condition.   1. **Design**   **4.1. UML diagrams**   1. Use case Diagram   Use case Diagram at its simplest is a representation of a user's interaction with the system that shows the relationship between the user and the different [use cases](https://en.wikipedia.org/wiki/Use_case) in which the user is involved. A use case diagram can identify the different types of users of a system and the different use cases and will often be accompanied by other types of diagrams as well. The use cases are represented by either circles or ellipses.    **Figure 4.1.1 Use Case diagram.**   1. **Class Diagram**   The class diagram is the main building block of [object-oriented](https://en.wikipedia.org/wiki/Object-oriented_programming) modeling. It is used for general [conceptual modeling](https://en.wikipedia.org/wiki/Conceptual_model) of the structure of the application, and for detailed modeling translating the models into [programming code](https://en.wikipedia.org/wiki/Programming_code). Class diagrams can also be used for [data modeling](https://en.wikipedia.org/wiki/Data_modeling).[[1]](https://en.wikipedia.org/wiki/Class_diagram#cite_note-1) The classes in a class diagram represent both the main elements, interactions in the application, and the classes to be programmed.  In the diagram, classes are represented with boxes that contain three compartments:   1. The top compartment contains the name of the class. It is printed in bold and centered, and the first letter is capitalized. 2. The middle compartment contains the attributes of the class. They are left-aligned and the first letter is lowercase. 3. The bottom compartment contains the operations the class can execute. They are also left-aligned and the first letter is lowercase.   In the design of a system, a number of classes are identified and grouped together in a class diagram that helps to determine the static relations between them. With detailed modeling, the classes of the conceptual design are often split into a number of subclasses.    **Figure 4.1.2 Class Diagram**   1. **Activity Diagram**   Activity diagram is another important diagram in UML to describe the dynamic aspects of the system. Activity diagram is basically a flowchart to represent the flow from one activity to another activity. The activity can be described as an operation of the system.  **Purpose of Activity Diagrams**  The basic purposes of activity diagrams is similar to other four diagrams. It captures the dynamic behavior of the system. Other four diagrams are used to show the message flow from one object to another but activity diagram is used to show message flow from one activity to another. Activity is a particular operation of the system. Activity diagrams are not only used for visualizing the dynamic nature of a system, but they are also used to construct the executable system by using forward and reverse engineering techniques. The only missing thing in the activity diagram is the message part.    **Figure.4.1.3 Activity Diagram**   1. **Sequence Diagram**   Sequence Diagrams are interaction diagrams that detail how operations are carried out. They capture the interaction between objects in the context of a collaboration. Sequence Diagrams are time focus and they show the order of the interaction visually by using the vertical axis of the diagram to represent time what messages are sent and when.  Purpose of Sequence Diagram   1. Model high-level interaction between active objects in a system 2. Model the interaction between object instances within a collaboration that realizes a use case 3. Model the interaction between objects within a collaboration that realizes an operation 4. Either model generic interactions (showing all possible paths through the interaction) or specific instances of a interaction (showing just one path through the interaction.     **Figure. 4.1.4 Sequence diagram**   1. **IMPLIMENTATION** |
|  |

* 1. **Module description**

**Esp32 Module.** It is a type of Node MCU module. It consists of inbuilt wifi and camera module. This module runs on a 3.3v to 5v. It is connected to any device using the wifi module and with it we can communicate with devices.

**Arduino.** It is a micro-controller and the Esp32 is programmed with the help of Arduino.

**L298N.** It is a motor driver which is connected with esp32 so as to deal with the directions provided by the user. It is also connected with the batteries for the power supply.

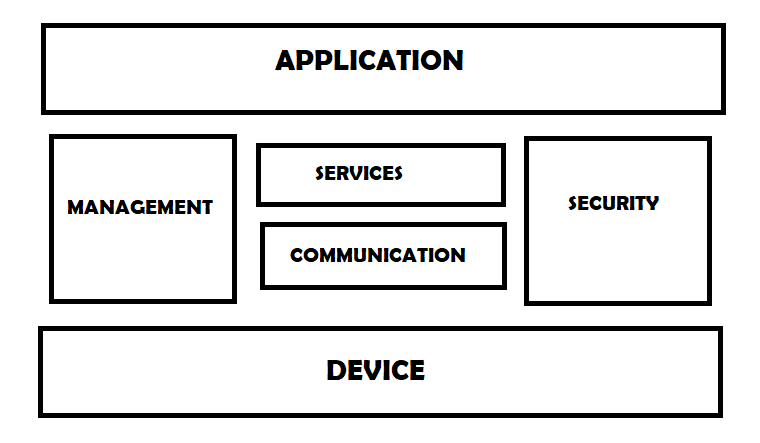
The UI has Video Streaming Module and Rover Control Module.

It has a GUI to control the rover. This control is achieved by connecting it over wifi. Buttons are used to control the rover in forward, backward, left or right directions.

1. **Main Module.** The process starts from the main module i.e to program ESP 32 module with the help of Arduino. All the other modules in this process are dependent on this module. It is like heart of this project.
2. **User-Interface Module.** It is responsible for operations which we perform at the user end. With the help of this we send commands to the rover so that it can act accordingly.
3. **Output Module.** In this we can see the data/information collected by the rover. It is responsible for displaying the live screening from the rover
   1. **Introduction of technologies used**

**INTERNET OF THINGS**

The Internet of Things (IoT) is the internetworking of physical devices, vehicles (connecting devices, and smart devices), buildings, and other items-including embedded with electronics, software’s, sensors, actuators, network connectivity that enable these objects to collect and exchange data. The IoT allows objects to be sensed or controlled remotely across existing network infrastructure, creating opportunities for more direct integration of the physical world into computer –based systems, and resulting in improved efficiency, accuracy and economic benefit in addition to reduced human intervention.

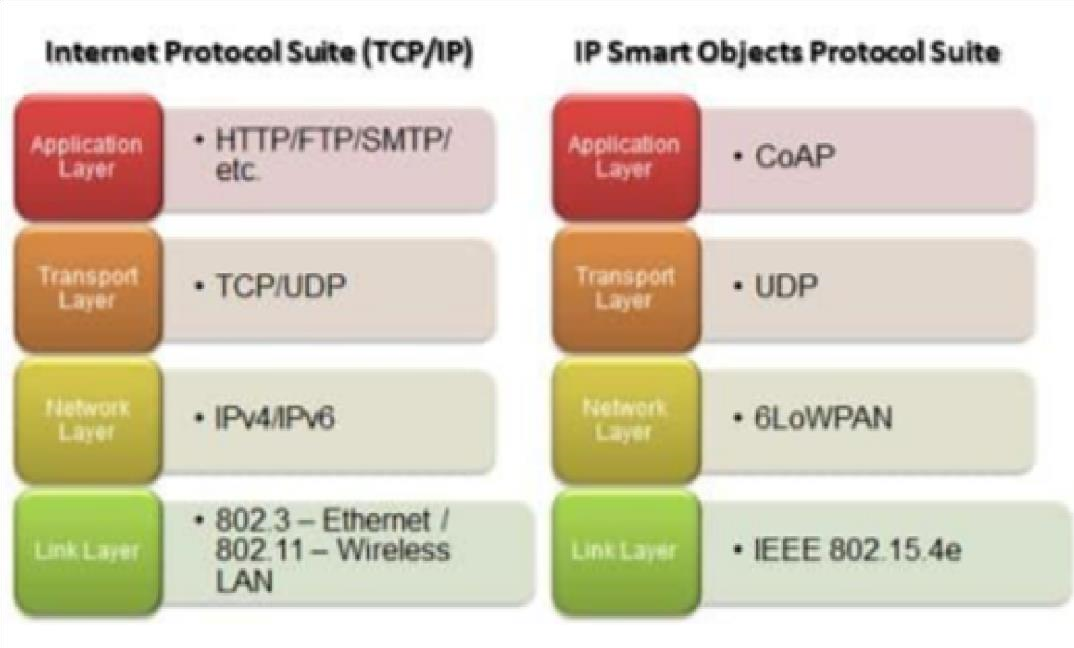
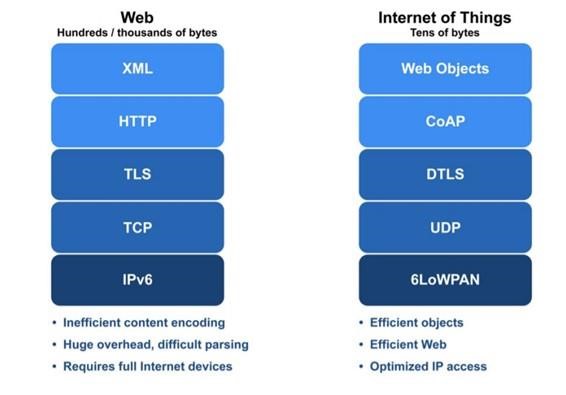


**Figure 5.1. IoT Functional Blocks**

An IoT system comprises a number of functional blocks that provide the system the capabilities for identification, sensing, actuation, and management as shown in figure. These functional blocks are described as follows:

1. Device: An IoT system comprises devices that provide sensing, actuation, monitoring and control functions.
2. Communication: The communication block handles the communication for the Iot system. The various protocols are used for communication by IoT systems
3. Services: An IoT system uses various types of IoT services such as services for device monitoring, device control services, data publishing services and services for device discovery.
4. Management: Management functional block provides various functions to govern the IoT system.
5. Security: Security functional block secures the IoT system and by providing functions such as authentication, authorization, message and content integrity, and data security.
6. Application: IoT applications provide an interface that the users can use to control and monitor various aspects of the IoT system. Applications also allow users to view the system status and view or analyse the processed data.

**IoT Protocols**



**Figure.5.2 IOT protocols.**

**Link Layer**  Link Layer protocols determine how the data is physically sent over the network's physical layer or medium (e. g., copper wire, coaxial cable, or a radio wave). The scope of the link layer is the network connection to which host is attached. Hosts on the same link exchange data packets over the link layer using link layer protocols. Link layer determines how the packets are coded and signalled by the hardware device over the medium to which the host is attached (such as a coaxial cable). Let us now look at some link layer protocols, which are relevant in the context of IoT.

1. **802.3-ETHERNET** IEEE 802.3 is a collection of wired Ethernet standards for the link layer. For example, 802.3 is the standard for 10BASIES Ethernet that uses coaxial cable as a shared medium, 802.3.i is the standard for 10BASE-T Ethernet over copper twisted-pair connections, 802.3ae is the standard for 10Gbit/Ethernet over fibre, and so on. optic connections, these standards provide data rates from 10Mbs to40Gbs and higher. The shared medium in Ethernet can be a coaxial cable, twisted-pair wire or an optical fibre. the shared medium (i.e., broadcast medium) carries the communication for all the devices propagation conditions and transceiver capabilities. The specifications of the 802.3 standards are available on the IEEE802.3 working group website.
2. **802.3-WIFI** IEEE802.11 is a collection of wireless local area network (WLAN)communication standards, including extensive description of the link layer. For example, 802.11a operates in the 5GHZ band, 802.11b and 802.11g operate in the 2.4GHZ band 802.11n operates in the 2.4/5GHZ bands, 802.11ac operates from 1MB/S to up to upto6.75Gb/s. The specifications of the 802.11 standards are available on the IEEE802.11 working group website
3. **802.11-WIFI** IEEE802.16 is a collection of wireless broadband standards, including extensive description for the link layer (also called WiMAX). WiMAX. Standards provide data rates from 1.5 Mb/s to 1 Gb/s. The recent update (802.16m) provides data rates of 100Mbit/s for mobile stations and 1 G bit/s for fixed stations. The specifications of the 802.11 standards are readily available on the IEEE802.16 working group website.
4. **802.15.4--LR-WPAN** IEEE 802.15.4 is a collection of standards for low-rate wireless personal area networks (LR-WPANs). These standards form the basis of specifications for high level communication protocols such as ZigBee. LRWPAN standards provide data rates from 4Kb/s 250 Kb/s. These standards provide low-cost and low-speed communication for power constrained devices. The specifications of The 802.15.4 standards are available on the IEEE802.15 working group website.
5. **2G/3/4G-MobileCommunication** There are different generations of mobile communication standards including second generation (2G including GSM and CDMA), third generation (3G-including UMTS and CDMA2000) and fourth generation (4G-including LTE). IOT devices based on these standards can communicate over cellular networks. Data rates for these standards range for 9.6 Kb/s(for 2G) to up to 100Mb/s(for 4G) and are available from the 3GPP websites.

**Network/Internet Layer:**

The network layers are responsible for sending IP datagrams from the source network to the destinations network. This layer performs the host addressing and packet routing. The data grams contain the source and destination addresses which are used to route them from the source to destination across multiple networks. Host identification is done using hierarchical IP addressing schemes such as IPV4 or IPV6.

**IPV4** Internet Protocol version 4(IPV4) is the most deployed Internet Protocol the is used to identify the devices on a network using a hierarchical addressing scheme. IPV4 uses a 32-bit address scheme that allows a total of 4,294,967,296 addresses. As more and more devices got connected to the Internet, these addresses got exhausted in the year 201. IPV4 has been succeeded by IPV6. The IP protocols establish connections on the packet network, but do not guarantee delivery of packets.

**IPV6** Internet Protocols version 6(IPV6) IS The newest version of Internet Protocols and successor to IPV4. IPV6 uses 128-bit address scheme that allows total of 3.4\*10 38 address.

**6LoWPAN** 6LoWPAN (IPV6 over Low power Wireless Personal Area Networks) brings IP protocol to the low-power devices which have limited processing capability 6LoWPAN works with the 802,15.4-based networks.

**Transport Layer** The transport layer protocols provide end-end message transfer capability independent of the underlying network. The message transfer capability can be set up on connections, either using handshakes (as in TCP) or without handshakes/acknowledgements (as in UDP). The transport layer provides functions such as error control, segmentation, flow control and congestion control.

1. **TCP** Transmission Control Protocol (TCP) is the most widely used transport layer protocol. that is used by web browsers along with HTTP, HTTPS application layer protocols, email programs (SMTP application layer protocols) and file transfer (FTP). TCP is connection oriented and stateful protocol. While IP protocol deals with sending packets. TCP ensures reliable transmission of packets in-order. TCP also provides error detection capability so that duplicate packets can be discarded and lost packets are retransmitted. The flow control capability of TCP ensures that the rate at which the sender sends the data is not too high for the receiver to process. The congestion collapse which can lead to degradation of network performance TCP is described in RFC 793.
2. **UDP** Unlike TCP, which requires carrying out an initial setup procedure. UDP of a connectionless protocol. UDP is useful for the time-sensitive applications that have very small data units to exchange and do not have stateless protocol. UDP does not provide guaranteed delivery or ensure connections created are reliable. UDP is described in RFC768.

**Application Layer** Application Layer protocols define how the applications interface with the lower layer protocols to send the data over the network. The application data, typically in files, is encoded by the application layer protocol and encapsulated in the transport layer protocol which provides connection or transaction-oriented communication over the network. Port numbers are used for application addresses (for example port 80 for HTTP port 22 for SSH etc.). Application layer protocols enable process-to-process connections using ports.

1. **HTTP** Hypertext Transfer Protocol (HTTP) is the application layer protocol that forms the foundation of the World Wide Web (WWW). HTTP includes commands such as GET, PUT, POST, DELETE, HEAD, TRACE, OPTIONS, etc. The protocol follows a request-response model where a client can send a request to a server using the HTTP commands. HTTP is a stateless protocol and each HTTP request is independent of the other requests. An HTTP client can be a browser or an application running on the client (e.g., an application running on an IOT device, a mobile application or other software).
2. **CoAP** Constrained Application Protocol (CoAP) is an application layer protocol for machine-to-machine(M2M) applications. meant for constrained environments with constrained devices and constrained networks. Like HTTP, CoAP is a web transfer protocol and uses a request-response model, however it runs on top of UDP instead of TCP. COAP uses a client-server architecture where clients communicate with servers using connectionless data grams COAP is designed to easily interface with HTTP. Like HTTP, COAP supports methods such as GET, PUT, POST, and DELETE, COAP draft specifications are available on IEFT Constrained environments (CORE) working Group Website.
3. **Web Socket** Web Socket protocol allows full-duplex communication over a single socket connection for sending messages between client and server. Web server is based on TCP and allows streams of messages to be sent back and forth between the client and server while keeping the TCP connection open. The client can be a browser, a mobile application or an IOT device.
4. **MQTT** Message Queue Telemetry Transport (MQTT) is a light-weight messaging protocol based on the publish-subscribe model. MQTT uses a clientserver (also called MQTT Broker) and publishes messages to topics. MQTT is well suited for constrained environments where the devices have limited processing and memory resources and the network bandwidth is low MQTT specifications are available on IBM developer Works.
5. **XMPP** Extensible Messaging and Presence Protocol (XMPP) is a protocol for real-time communication and streaming XML data between entities. XMPP powers a wide range of applications including messaging, presence, data syndication, gaming, multi-party chat and voice/video calls. XMPP allows sending small chunks of XML data from one network entry to another in near real-time. XMPP is a decentralized protocol and uses a client-server architecture. XMPP supports both client-to-server and server-to-server communication paths. In the context of IOT, XMPP allows real-time communication between IOT devices.
6. **DDS** Data Distribution Service (DDS) is a data-centric middleware standard for device to-device or machine-to-machine. DDS uses a publish-subscribe for models where publishers (e g. devices that generate data) create topics to which subscribers (e.g., devices that want to consume data) can subscribe. Publisher is an object responsible for data distribution and the subscriber is responsible for receiving reliability. DDS provides quality-of-service (QOS) control and configurable reliability .DDS is described in Object Management Group (OMG)DDS specification.
7. **AMQP** Advanced Messaging Queuing Protocol (AMQP) is an open application layer protocol for business messaging. AMQP supports both point-to-point publishers (e.g devices or applications that generate data). Publishers publish the connections to consumers (applications that process data). Publishers publish the messages to exchange which then distribute message copies to queues. Messages are either delivered by the broker to the consumers which have subscribed to the queues or the consumers can pull the messages from the queues. AMQP specification is available on the AMQP working group website.

**MACHINE LEARNING**

The term Machine Learning was coined by Arthur Samuel in 1959, an American pioneer in the field of computer gaming and artificial intelligence, and stated that “it gives computers the ability to learn without being explicitly programmed”.   
And in 1997, Tom Mitchell gave a “well-posed” mathematical and relational definition that “A computer program is said to learn from experience E with respect to some task T and some performance measure P, if its performance on T, as measured by P, improves with experience E.

Machine Learning is the latest buzzword floating around. It deserves to, as it is one of the most interesting subfields of Computer Science. So what does Machine Learning really mean?

Let’s try to understand Machine Learning in layman’s terms. Consider you are trying to toss a paper into a dustbin.

After the first attempt, you realize that you have put too much force into it. After the second attempt, you realize you are closer to the target but you need to increase your throw angle. What is happening here is basically after every throw we are learning something and improving the end result. We are programmed to learn from our experience.

This implies that the tasks in which machine learning is concerned to offer a fundamentally operational definition rather than defining the field in cognitive terms. This follows Alan Turing’s proposal in his paper “Computing Machinery and Intelligence”, in which the question “Can machines think?” is replaced with the question “Can machines do what we (as thinking entities) can do?”   
Within the field of data analytics, machine learning is used to devise complex models and algorithms that lend themselves to prediction; in commercial use, this is known as predictive analytics. These analytical models allow researchers, data scientists, engineers, and analysts to “produce reliable, repeatable decisions and results” and uncover “hidden insights” through learning from historical relationships and trends in the data set(input).

**MACHINE LEARNING**

The term Machine Learning was coined by Arthur Samuel in 1959, an American pioneer in the field of computer gaming and artificial intelligence, and stated that “it gives computers the ability to learn without being explicitly programmed”.

And in 1997, Tom Mitchell gave a “well-posed” mathematical and relational definition that “A computer program is said to learn from experience E with respect to some task T and some performance measure P, if its performance on T, as measured by P, improves with experience E.

Machine Learning is the latest buzzword floating around. It deserves to, as it is one of the most interesting subfields of Computer Science. So what does Machine Learning really mean?

Let’s try to understand Machine Learning in layman’s terms. Consider you are trying to toss a paper into a dustbin.

After the first attempt, you realize that you have put too much force into it. After the second attempt, you realize you are closer to the target but you need to increase your throw angle. What is happening here is basically after every throw we are learning something and improving the end result. We are programmed to learn from our experience.

This implies that the tasks in which machine learning is concerned to offer a fundamentally operational definition rather than defining the field in cognitive terms. This follows Alan Turing’s proposal in his paper “Computing Machinery and Intelligence”, in which the question “Can machines think?” is replaced with the question “Can machines do what we (as thinking entities) can do?”

Within the field of data analytics, machine learning is used to devise complex models and algorithms that lend themselves to prediction; in commercial use, this is known as predictive analytics. These analytical models allow researchers, data scientists, engineers, and analysts to “produce reliable, repeatable decisions and results” and uncover “hidden insights” through learning from historical relationships and trends in the data set(input).

**Classification of Machine Learning**

Machine learning implementations are classified into three major categories, depending on the nature of the learning “signal” or “response” available to a learning system which is as follows:-

Supervised learning: When an algorithm learns from example data and associated target responses that can consist of numeric values or string labels, such as classes or tags, in order to later predict the correct response when posed with new examples comes under the category of Supervised learning. This approach is indeed similar to human learning under the supervision of a teacher. The teacher provides good examples for the student to memorize, and the student then derives general rules from these specific examples.

Unsupervised learning: Whereas when an algorithm learns from plain examples without any associated response, leaving to the algorithm to determine the data patterns on its own. This type of algorithm tends to restructure the data into something else, such as new features that may represent a class or a new series of un-correlated values. They are quite useful in providing humans with insights into the meaning of data and new useful inputs to supervised machine learning algorithms.

As a kind of learning, it resembles the methods humans use to figure out that certain objects or events are from the same class, such as by observing the degree of similarity between objects. Some recommendation systems that you find on the web in the form of marketing automation are based on this type of learning.

Reinforcement learning: When you present the algorithm with examples that lack labels, as in unsupervised learning. However, you can accompany an example with positive or negative feedback according to the solution the algorithm proposes comes under the category of Reinforcement learning, which is connected to applications for which the algorithm must make decisions (so the product is prescriptive, not just descriptive, as in unsupervised learning), and the decisions bear consequences. In the human world, it is just like learning by trial and error.

Errors help you learn because they have a penalty added (cost, loss of time, regret, pain, and so on), teaching you that a certain course of action is less likely to succeed than others. An interesting example of reinforcement learning occurs when computers learn to play video games by themselves.

In this case, an application presents the algorithm with examples of specific situations, such as having the gamer stuck in a maze while avoiding an enemy. The application lets the algorithm know the outcome of actions it takes, and learning occurs while trying to avoid what it discovers to be dangerous and to pursue survival. You can have a look at how the company Google DeepMind has created a reinforcement learning program that plays old Atari’s video games. When watching the video, notice how the program is initially clumsy and unskilled but steadily improves with training until it becomes a champion.

Semi-supervised learning: where an incomplete training signal is given: a training set with some (often many) of the target outputs missing. There is a special case of this principle known as Transduction where the entire set of problem instances is known at learning time, except that part of the targets are missing.

**Categorizing on the basis of required Output**

Another categorization of machine learning tasks arises when one considers the desired output of a machine-learned system:

**Classification:** When inputs are divided into two or more classes, and the learner must produce a model that assigns unseen inputs to one or more (multi-label classification) of these classes. This is typically tackled in a supervised way. Spam filtering is an example of classification, where the inputs are email (or other) messages and the classes are “spam” and “not spam”.

**Regression:** Which is also a supervised problem, A case when the outputs are continuous rather than discrete.

**Clustering:** When a set of inputs is to be divided into groups. Unlike in classification, the groups are not known beforehand, making this typically an unsupervised task.

Machine Learning comes into the picture when problems cannot be solved by means of typical approaches.

**Applications**

The applications of Internet of Things span a wide range of domains including (but not limited to) homes, cities, environment, energy, systems, retail, logistics, industry, agriculture and health as listed. Machine learning (**ML**) is a type of artificial intelligence (AI) that allows software applications to become more accurate at predicting outcomes without being explicitly programmed to do so. Machine learning algorithms use historical data as input to predict new output values

1. For homes, IoT has several applications such as smart lighting that adapt the lighting to suit the ambient conditions. smart appliances that can be remotely monitored and controlled, intrusion detection systems, smart smoke detectors etc.,
2. For cities, IoT has applications such as smart parking systems that provide status updates on available slots, smart lighting that helps in saving energy, smart roads that provide information on driving conditions and structural health monitoring systems.
3. For the environment, IoT has applications such as weather monitoring, air and noise pollution, forest fire detection and river flood detection systems.
4. For energy systems, IoT has applications such as including smart grids, grid integration of renewable energy sources and prognostic health management systems.
5. For the retail domain, IoT has applications such as inventory management, smart payments and smart vending machines.
6. For the agriculture domain, IoT has applications such as smart irrigation systems that help in saving water while enhancing productivity and greenhouse control systems.
7. Industrial applications such as smart irrigation systems. Industrial applications of IoT include machine diagnostics and prognosis systems.
8. For health and lifestyle, IoT has applications such as health and fitness monitoring systems and wearable electronics.

**5.4. Sample Code**

#include <Arduino.h>

#include <WiFi.h>

#include <WiFiClientSecure.h>

#include "soc/soc.h"

#include "soc/rtc\_cntl\_reg.h"

#include "esp\_camera.h"

#include <UniversalTelegramBot.h>

#include <ArduinoJson.h>

const char\* ssid = "SATYA111";

const char\* password = "satya111";

// Initialize Telegram BOT

StringBOTtoken = "1760483053:AAEBEeLKesH1wVaqOLhqjJW7wGMxOY94d3I"; // your Bot Token (Get from Botfather)

// Use @myidbot to find out the chat ID of an individual or a group

// Also note that you need to click "start" on a bot before it can

// message you

String CHAT\_ID = "980002976";

bool sendPhoto = false;

WiFiClientSecure clientTCP;

UniversalTelegramBot bot(BOTtoken, clientTCP);

#define FLASH\_LED\_PIN 4

bool flashState = LOW;

//Checks for new messages every 1 second.

int botRequestDelay = 1000;

unsigned long lastTimeBotRan;

//CAMERA\_MODEL\_AI\_THINKER

#define PWDN\_GPIO\_NUM 32

#define RESET\_GPIO\_NUM -1

#define XCLK\_GPIO\_NUM 0

#define SIOD\_GPIO\_NUM 26

#define SIOC\_GPIO\_NUM 27

#define Y9\_GPIO\_NUM 35

#define Y8\_GPIO\_NUM 34

#define Y7\_GPIO\_NUM 39

#define Y6\_GPIO\_NUM 36

#define Y5\_GPIO\_NUM 21

#define Y4\_GPIO\_NUM 19

#define Y3\_GPIO\_NUM 18

#define Y2\_GPIO\_NUM 5

#define VSYNC\_GPIO\_NUM 25

#define HREF\_GPIO\_NUM 23

#define PCLK\_GPIO\_NUM 22

void configInitCamera(){

camera\_config\_t config;

config.ledc\_channel = LEDC\_CHANNEL\_0;

config.ledc\_timer = LEDC\_TIMER\_0;

config.pin\_d0 = Y2\_GPIO\_NUM;

config.pin\_d1 = Y3\_GPIO\_NUM;

config.pin\_d2 = Y4\_GPIO\_NUM;

config.pin\_d3 = Y5\_GPIO\_NUM;

config.pin\_d4 = Y6\_GPIO\_NUM;

config.pin\_d5 = Y7\_GPIO\_NUM;

config.pin\_d6 = Y8\_GPIO\_NUM;

config.pin\_d7 = Y9\_GPIO\_NUM;

config.pin\_xclk = XCLK\_GPIO\_NUM;

config.pin\_pclk = PCLK\_GPIO\_NUM;

config.pin\_vsync = VSYNC\_GPIO\_NUM;

config.pin\_href = HREF\_GPIO\_NUM;

config.pin\_sscb\_sda = SIOD\_GPIO\_NUM;

config.pin\_sscb\_scl = SIOC\_GPIO\_NUM;

config.pin\_pwdn = PWDN\_GPIO\_NUM;

config.pin\_reset = RESET\_GPIO\_NUM;

config.xclk\_freq\_hz = 20000000;

config.pixel\_format = PIXFORMAT\_JPEG;

//init with high specs to pre-allocate larger buffers

if(psramFound()){

config.frame\_size = FRAMESIZE\_UXGA;

config.jpeg\_quality = 10; //0-63 lower number means higher quality

config.fb\_count = 2;

} else {

config.frame\_size = FRAMESIZE\_SVGA;

config.jpeg\_quality = 12; //0-63 lower number means higher quality

config.fb\_count = 1;

}

// camera init

esp\_err\_t err = esp\_camera\_init(&config);

if (err != ESP\_OK) {

Serial.printf("Camera init failed with error 0x%x", err);

delay(1000);

ESP.restart();

}

// Drop down frame size for higher initial frame rate

sensor\_t \* s = esp\_camera\_sensor\_get();

s->set\_framesize(s, FRAMESIZE\_CIF); // UXGA|SXGA|XGA|SVGA|VGA|CIF|QVGA|HQVGA|QQVGA

}

void handleNewMessages(int numNewMessages) {

Serial.print("Handle New Messages: ");

Serial.println(numNewMessages);

for (int i = 0; i < numNewMessages; i++) {

String chat\_id = String(bot.messages[i].chat\_id);

if (chat\_id != CHAT\_ID){

bot.sendMessage(chat\_id, "Unauthorized user", "");

continue;

}

// Print the received message

String text = bot.messages[i].text;

Serial.println(text);

String from\_name = bot.messages[i].from\_name;

if (text == "/start") {

String welcome = "Welcome , " + from\_name + "\n";

welcome += "Use the following commands to interact with the ESP32-CAM \n";

welcome += "/photo : takes a new photo\n";

welcome += "/flash : toggles flash LED \n";

bot.sendMessage(CHAT\_ID, welcome, "");

}

if (text == "/flash") {

flashState = !flashState;

digitalWrite(FLASH\_LED\_PIN, flashState);

Serial.println("Change flash LED state");

}

if (text == "/photo") {

sendPhoto = true;

Serial.println("New photo request");

}

}

}

String sendPhotoTelegram() {

const char\* myDomain = "api.telegram.org";

String getAll = "";

String getBody = "";

camera\_fb\_t \* fb = NULL;

fb = esp\_camera\_fb\_get();

if(!fb) {

Serial.println("Camera capture failed");

delay(1000);

ESP.restart();

return "Camera capture failed";

}

Serial.println("Connect to " + String(myDomain));

if (clientTCP.connect(myDomain, 443)) {

Serial.println("Connection successful");

String head = "--RandomNerdTutorials\r\nContent-Disposition: form-data; name=\"chat\_id\"; \r\n\r\n" + CHAT\_ID + "\r\n--RandomNerdTutorials\r\nContent-Disposition: form-data; name=\"photo\"; filename=\"esp32-cam.jpg\"\r\nContent-Type: image/jpeg\r\n\r\n";

String tail = "\r\n--RandomNerdTutorials--\r\n";

uint16\_t imageLen = fb->len;

uint16\_t extraLen = head.length() + tail.length();

uint16\_t totalLen = imageLen + extraLen;

clientTCP.println("POST /bot"+BOTtoken+"/sendPhoto HTTP/1.1");

clientTCP.println("Host: " + String(myDomain));

clientTCP.println("Content-Length: " + String(totalLen));

clientTCP.println("Content-Type: multipart/form-data; boundary=RandomNerdTutorials");

clientTCP.println();

clientTCP.print(head);

uint8\_t \*fbBuf = fb->buf;

size\_t fbLen = fb->len;

for (size\_t n=0;n<fbLen;n=n+1024) {

if (n+1024<fbLen) {

clientTCP.write(fbBuf, 1024);

fbBuf += 1024;

}

else if (fbLen%1024>0) {

size\_t remainder = fbLen%1024;

clientTCP.write(fbBuf, remainder);

}

}

clientTCP.print(tail);

esp\_camera\_fb\_return(fb);

int waitTime = 10000; // timeout 10 seconds

long startTimer = millis();

boolean state = false;

while ((startTimer + waitTime) > millis()){

Serial.print(".");

delay(100);

while (clientTCP.available()) {

char c = clientTCP.read();

if (state==true) getBody += String(c);

if (c == '\n') {

if (getAll.length()==0) state=true;

getAll = "";

}

else if (c != '\r')

getAll += String(c);

startTimer = millis();

}

if (getBody.length()>0) break;

}

clientTCP.stop();

Serial.println(getBody);

}

else {

getBody="Connected to api.telegram.org failed.";

Serial.println("Connected to api.telegram.org failed.");

}

return getBody;

}

void setup(){

WRITE\_PERI\_REG(RTC\_CNTL\_BROWN\_OUT\_REG, 0);

// Init Serial Monitor

Serial.begin(115200);

// Set LED Flash as output

pinMode(FLASH\_LED\_PIN, OUTPUT);

digitalWrite(FLASH\_LED\_PIN, flashState);

// Config and init the camera

configInitCamera();

// Connect to Wi-Fi

WiFi.mode(WIFI\_STA);

Serial.println();

Serial.print("Connecting to ");

Serial.println(ssid);

WiFi.begin(ssid, password);

clientTCP.setCACert(TELEGRAM\_CERTIFICATE\_ROOT); // Add root certificate for api.telegram.org

while (WiFi.status() != WL\_CONNECTED) {

Serial.print(".");

delay(500);

}

Serial.println();

Serial.print("ESP32-CAM IP Address: ");

Serial.println(WiFi.localIP());

}

void loop() {

if (sendPhoto) {

Serial.println("Preparing photo");

sendPhotoTelegram();

sendPhoto = false;

}

if (millis() > lastTimeBotRan + botRequestDelay) {

int numNewMessages = bot.getUpdates(bot.last\_message\_received + 1);

while (numNewMessages) {

Serial.println("got response");

handleNewMessages(numNewMessages);

numNewMessages = bot.getUpdates(bot.last\_message\_received + 1);

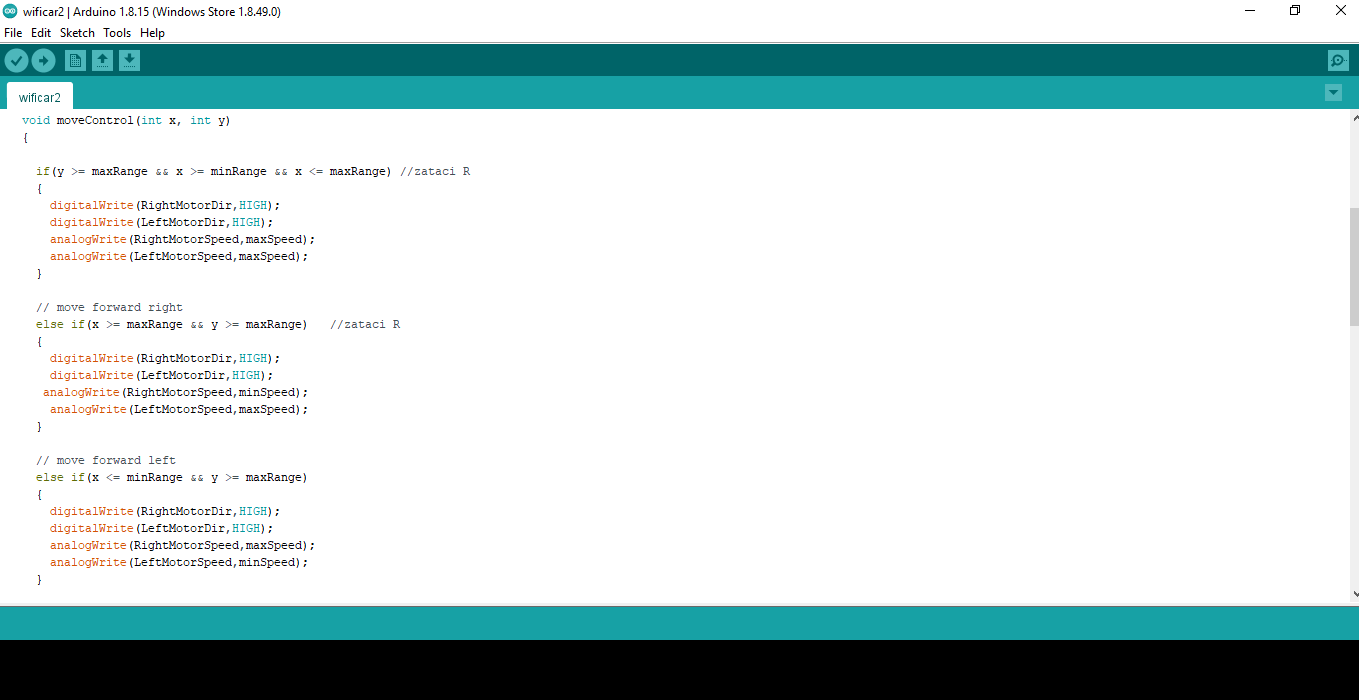
}

lastTimeBotRan = millis();

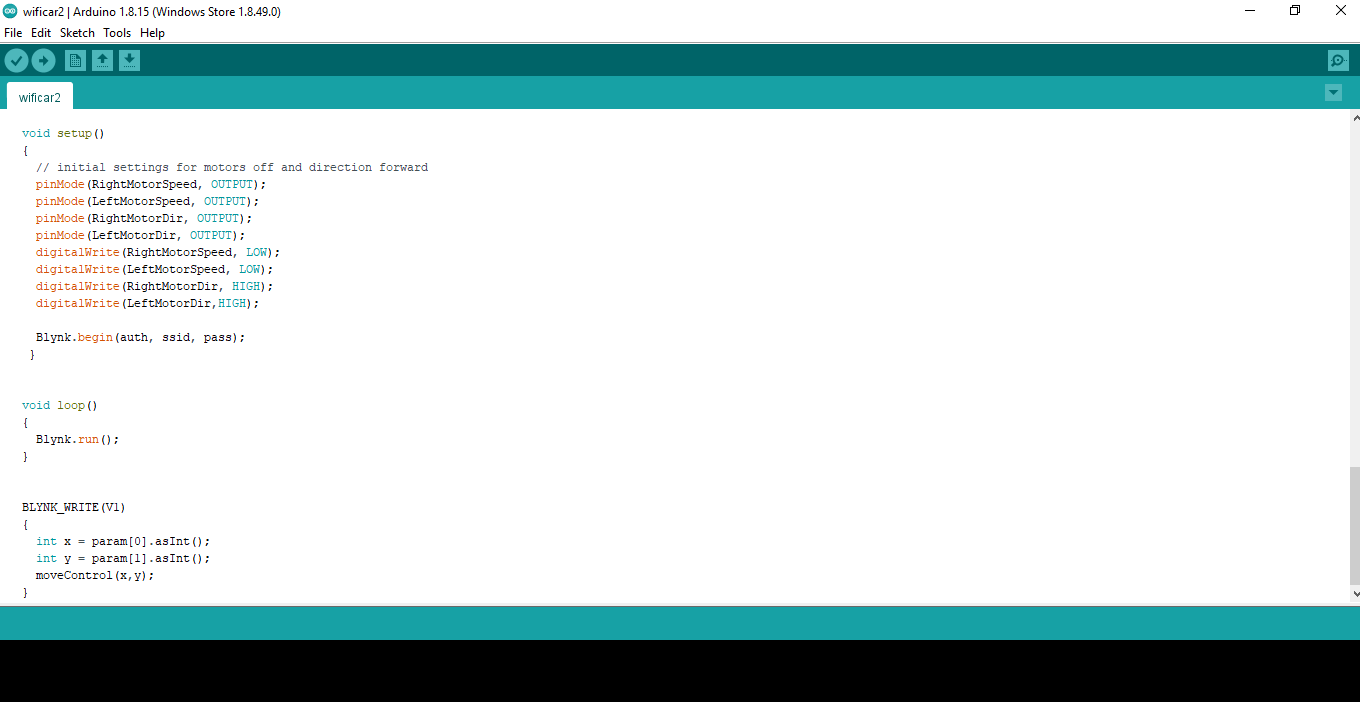
}

}

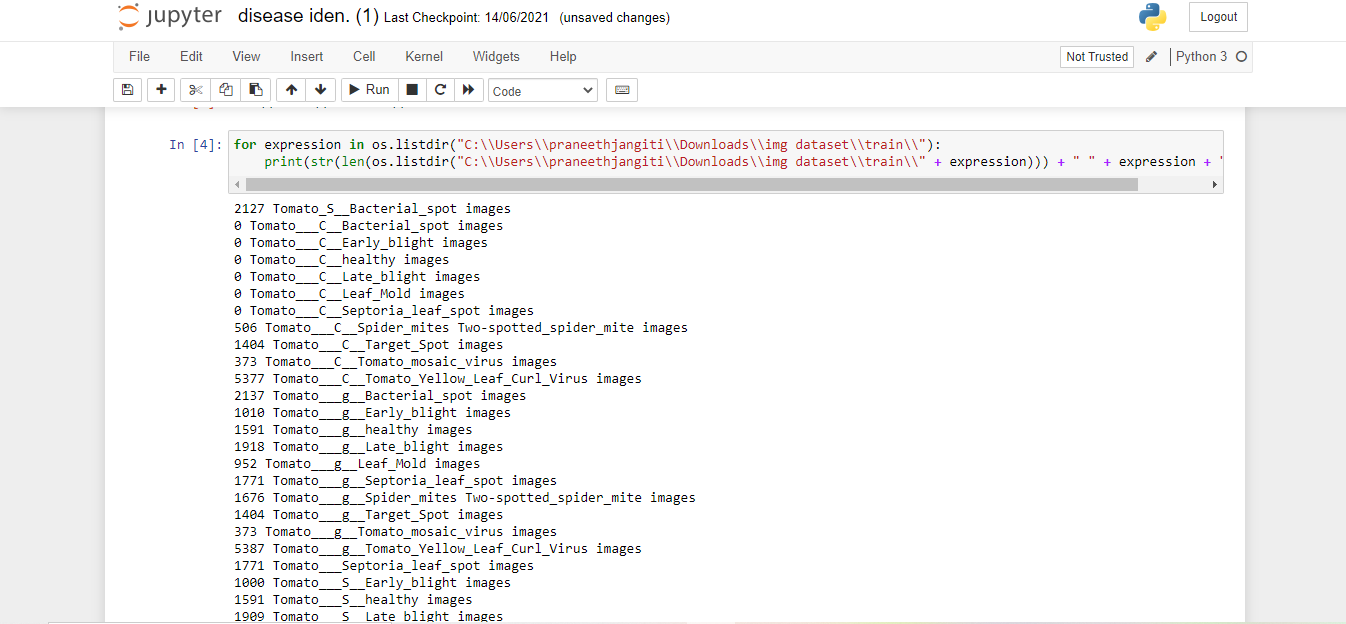
1. **TESTCASES**

****

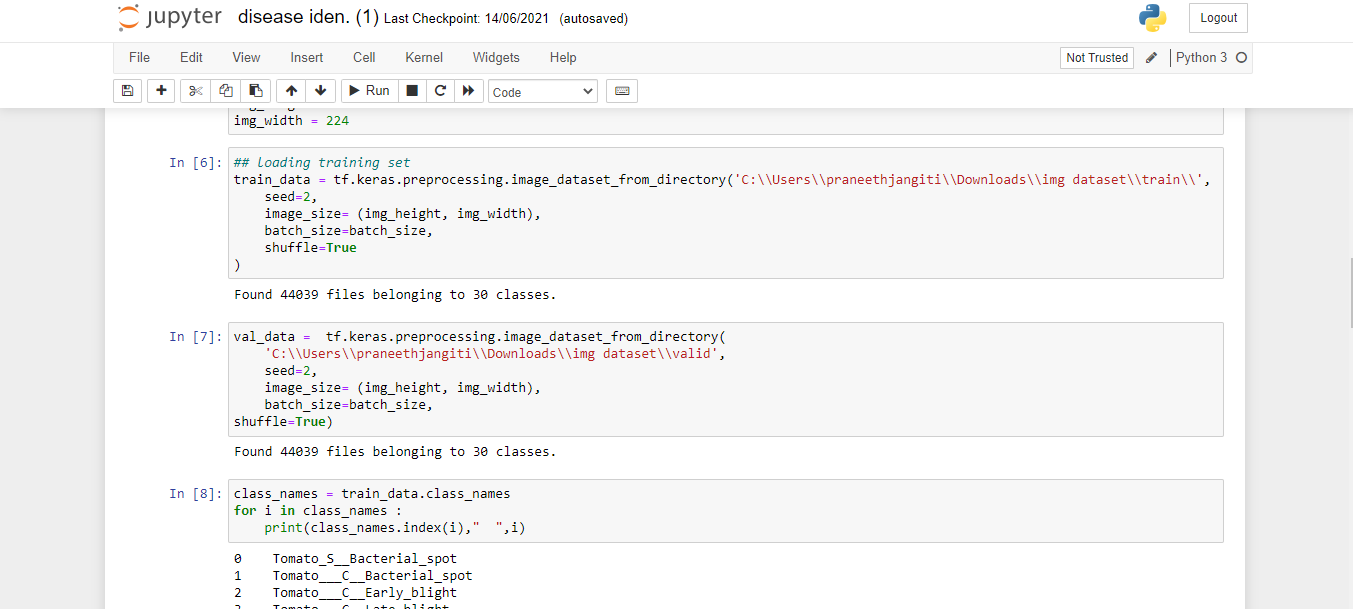
**Figure 6.1**. The above picture shows the code of rover movement.



**Figure 6.2.**  Blynk server establishment with the rover .

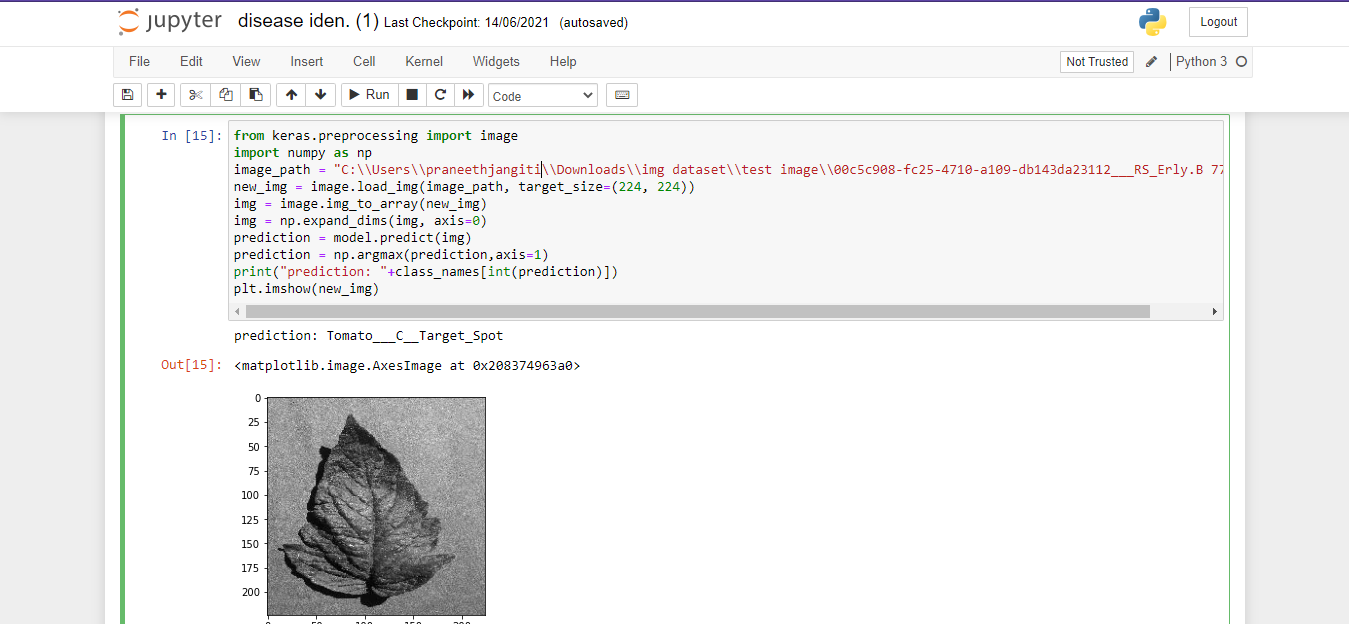


**Figure 6.3** The above picture shows different classes into which the tomato crop images are classified.



**Figure 6.4** In this the images folder is uploaded to the code using the path i.e the location of the folder.

1. **SREENSHOTS**

****

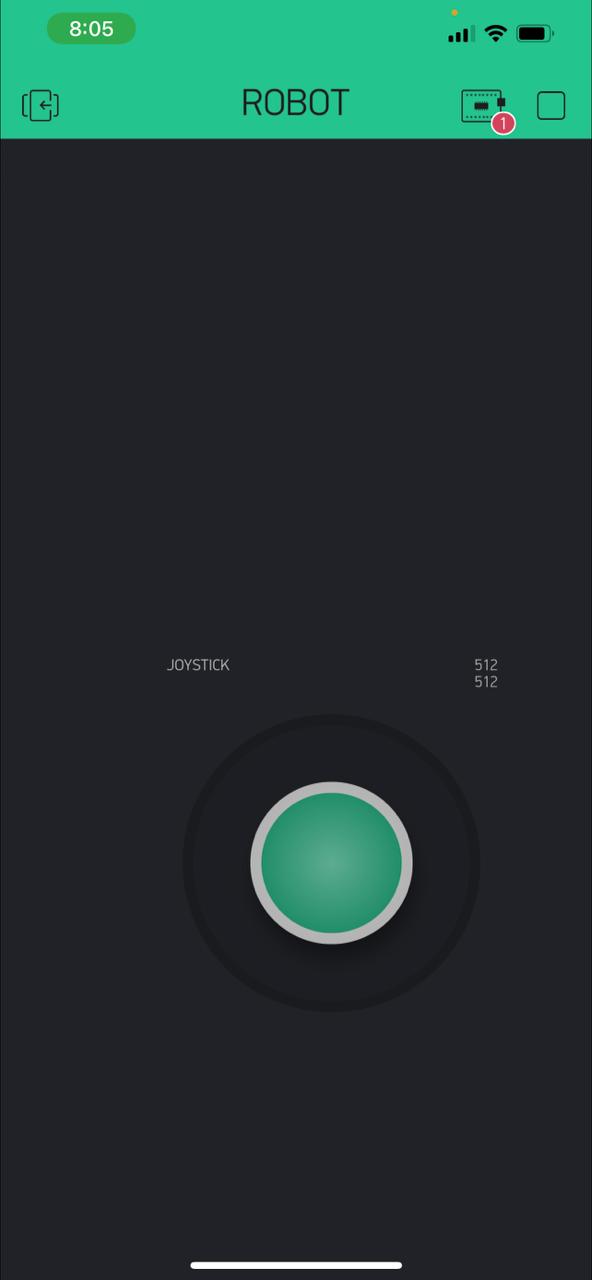
**Screenshot 7.1. Predicting the test image**

The picture shows that output of the model i.e it shows the predicted class of the test image.



**Screenshot 7.2. Photo capture bot**

The above screenshot displays the interface of the photo capture bot displaying the commands given to the bot by the user and the output image given by the bot to the user.



**Screenshot 7.3 User Interface**

This is the User-Interface through which we operate the rover. The above Figure displays the interface for movement of robotic car as shown in figure which is controlled via phone/Desktop.



**Screenshot 7.4. Front view of the Rover**

In the above picture we have the front view of the rover which has Mi camera and ESP32 module.



**Screenshot 7.5. Rover**

Rover with stand and to the stand the ESP32 is attached so that the upper part of the crop can also be captured.



**Screenshot 7.5. MI Camera view**

The above screenshot displays the view of the mi Camera that is showing live stream while the rover is moving. It acts as the eye to the rover.

1. **CONCLUSION**

A rover platform is designed, built and programmed for agricultural applications. It is developed as a fully autonomous rover by adding camera and various sensors and digital camera to have better approach for agricultural applications. So, it will be able to monitor the tomato crops in any given location without any human interaction and control. Data transfer will be achieved by wireless communication instead of using serial cable as before. Damaged tomato crops can be identified using this Tomato Crop Monitoring via Rover.

1. **Future Enhancement**

This project offers a lot of scope for adding newer features. Since all images are captured using telegram bot, there are no resource constraints apart from using an application to capture images. We can program the rover such that it can detect diseased crop and capture those images of crop on its own. Thus, we can make it completely autonomous. Also, with the presence of GPS navigation and mapping software, the rover has the capability of finding the best route possible to reach a certain location. Also, here we can use cloud instead of storing the captured images locally and from the cloud directly we can use them for later research .we can make it an allterrain robot, which would make it ideal for a surveillance rover. There is also the option of adding sound processing to the remote computer, thus giving it greater surveillance capabilities. The possibilities are endless. This rover in its current state provides a platform for further research into improving its capabilities.

**BIBLIOGRAPHY**

[1] <https://www.ijrte.org/wp-content/uploads/papers/v7i6s3/F1041376S19.pdf>

[2] SwapnilLagad, SwapniliKarmore, “Algorithms of Wheat Disease Identification in Spraying Robot System”, IEEE-2017.

[3] ZhihuaDiao, ChunyingDiao, Yuanyuan Wu, “Algorithms of Wheat Disease Identification in Spraying Robot System”, IEEE-2017.

[4] Ronberestein,YaelEdan, “Automatic Adjustable Spraying Device for Site-Specific Agricultural Application”, IEEE-2018.

[5] Chun-Mu Wu ,Jui-Tsung Lu,” : Implementation of remote control for a spraying robot”, IEEE-2017.

[6] I.F. Akyildiz, W. Su, Y. Sankarasubramaniam, E. Cayirci, A survey on sensor networks, IEEE Communications Magazine 40 (8) (2002) 104–112.

[7] L. Atzori, A. Iera, and G. Morabito, "The Internet of Things: A survey," Computer Networks, vol. 54, pp. 2787-2805, 10/28/ 2010.

[8] H. Chang, N. Zhou, X. Zhao, Q. Cao, M. Tan and Y. Zhang, "A new agriculture monitoring system based on WSNs," 2014 12th International Conference on Signal Processing (ICSP), Hangzhou, 2014, pp. 1755-1760.

[9] Dr. AntoBennet, M, SankarBabu G, Natarajan S, “Reverse Room Techniques for Irreversible Data Hiding”, Journal of Chemical and Pharmaceutical Sciences 08(03): 469-475, September 2015.

[10] Dr. AntoBennet, M ,Sankaranarayanan S, SankarBabu G, “ Performance & Analysis of Effective Iris Recognition System Using Independent Component Analysis”, Journal of Chemical and Pharmaceutical Sciences 08(03): 571-576, August 2015.

[11] Dr. AntoBennet, M, Suresh R, Mohamed Sulaiman S, “Performance &analysis of automated removal of head movement artifacts in EEG using brain computer interface”, Journal of Chemical and Pharmaceutical Research 07(08): 291-299, August 2015.

[12] Dr. AntoBennet, M “A Novel Effective Refined Histogram For Supervised Texure Classification”, International Journal of Computer & Modern Technology , Issue 01 ,Volume02 ,pp 67-73, June 2015.

[13] Dr. AntoBennet, M, SrinathR,RaishaBanuA,“Development of Deblocking Architectures for block artifact reduction in videos”, International Journal of Applied Engineering Research,Volume 10, Number 09 (2015) pp. 6985-6991, April 2015.

[14] Rajesh, M., and J. M. Gnanasekar. "Path Observation Based Physical Routing Protocol for Wireless Ad Hoc Networks." Wireless Personal Communications 97.1 (2017): 1267- 1289.

[15] Rajesh, M., and J. M. Gnanasekar. "Sector Routing Protocol (SRP) in Ad-hoc Networks." Control Network and Complex Systems 5.7 (2015): 1-4.

[16] Rajesh, M. "A Review on Excellence Analysis of Relationship Spur Advance in Wireless Ad Hoc Networks." International Journal of Pure and Applied Mathematics 118.9 (2018): 407-412.

[17] Rajesh, M., et al. "SENSITIVE DATA SECURITY IN CLOUD COMPUTING AID OF DIFFERENT ENCRYPTION TECHNIQUES." Journal of Advanced Research in Dynamical and Control Systems .

[18] Rajesh, M. "A signature based information security system for vitality proficient information accumulation in wireless sensor .

[19] systems." International Journal of Pure and Applied Mathematics 118.9 (2018): 367-387. [20] Rajesh, M., K. Balasubramaniaswamy, and S. Aravindh. "MEBCK from Web using NLP Techniques." Computer Engineering and Intelligent Systems 6.8: 24-26.