Iot Based Surveillance Robot Using Machine Learning for Rescue Operation.

1st Md Shahriar Hossain Apu
Program IoT, Dept. of ICT
Bangabandhu Sheikh Mujibur Rahman Digital University
Gazipur, Bangladesh
1901036@iot.bdu.ac,bd

2nd Chowdhury Rafsan Mahmudllah
Program IoT, Dept. of ICT
Bangabandhu Sheikh Mujibur Rahman Digital University
Gazipur, Bangladesh
1801048@iot.bdu.ac,bd

3rd Tanvir Hasan
Program IoT, Dept. of ICT
Bangabandhu Sheikh Mujibur Rahman Digital University
Gazipur, Bangladesh
1901009@iot.bdu.ac,bd

4th Sadia Hossain
Program IoT, Dept. of ICT
Bangabandhu Sheikh Mujibur Rahman Digital University
Gazipur, Bangladesh
1901039@iot.bdu.ac,bd

***Abstract*—Practically speaking, robots are built to perform special functions that cannot be handled by humans, or where conditions for humans to work are not certain and risky. Thus, these types of vehicles can perform the role that is difficult for humans. This motivated us to create a robot capable of performing risky tasks like border patrol and surveillance. The main purpose of this vehicle is to be controlled using wireless communication technology and travel long distance without any problem. This paper focuses on bringing together various technologies such as IOT, Software, Wireless communication, and mobile application. It uses wifi and Bluetooth for communication and tracking the vehicle, Python/OpenCV with object detection and ESP32 Cam to find the enemies and obstacles. This paper presents a low-cost human-computer interaction device represented by Arduino control Bluetooth car.A object detection Python/OpenCV program is used to detect and track the moment of the object. A mobile application is also used to control the vehicle and provide a secondary manual control.**

***Index Terms*—ESP32 CAM, TensorFlow, Object Detection, Opencv, IoT, Machine Learning,Bluetooth-Car, YOLOv5**

I. Introduction

With the increasing trend of making devices wireless and user friendly, this paper focuses on bringing this ease of control into surveillance vehicle. The vehicle is intended to be controlled by Bluetooth technology With the assistance of wireless communication, connecting with the vehicle in a more user-friendly manner than the normal manual-controlled vehicle is simpler. The overall structure of vehicle involves Arduino Uno and HC06 Bluetooth module. The movement of the vehicle can be in four directions that are left, right, forward and backward. To avoid unwanted movement of vehicle due to small movement of hand, we have set a dead-zone where small movement will not be tracked. There is a camera mounted on vehicle and its live feed is broadcasted to the user. The live video captured by the camera will be sent to the screen by which we will detect the object. Camera used in the vehicle is an ESP32 CAM which will detect and recognize the object using Artificial Intelligence. The objects appearing in front of the vehicle will be notified to the user denoting the distance from the object and it will be tracked through camera using python/OpenCV program. OpenCV is an open-sourced image processing library that is very widely used not just in industry but also in the field of research and development. Here for object detection cvlib Library is used. The library uses a pretrained AI model on the COCO dataset to detect objects. The name of the pre-trained model is YOLO v5. For hardware, the ESP32 Camera Module was used which can be programmed through FTDI Module. It is required to set up the Arduino IDE for the ESP32 Camera Module. It is important to upload the firmware and then work on the object detection identification part. The script for object detection is written in the python programming language, thus people who want to use this code will also have to install Python and its required Libraries. This project makes use of OpenCV for Object Detection and Identification.

The applications and widespread use of machine learning algorithms have made a significant change in the way we perceive computer vision problems. With the introduction of deep learning into the field of image classification, the dynamics of real-time object detection have faced a great impact. In deep learning, the mapping is done by using representation-learning algorithms. These representations are expressed in terms of other, simpler representations. In other words, a deep learning system can represent the concept of an image for an object by combining simpler concepts, such as points and lines, which are in turn defined in terms of edges. By using a variety of algorithms, a bench marking dataset and correct labeling packages a system can be trained to achieve the desired output. A fundamental aspect of deep learning in image classification is the use of Convolutions architectures.

II. METHODS AND MATERIALS

A. PROPOSED SYSTEM

The fundamental necessity of any robot is its structure or its body on which its complete control circuitry and actuators are to be mounted. Our primary objective in our design is to move the unit according to the command given by the user. So, we designed a simple robot which can move forward, backward, turn right and left according to the command. The ESP-32 CAM can track the vehicles/objects in its view and recognize the object. Both the sections sender and receiver communicate with each other through the HC-06 Module which has a range of approx. 100 meter. Before deciding the direction of the robotic vehicle, we capture the continuous stream of video from camera in the vehicle to our customized application. The signals at both ends are decoded and encoded by the respective Arduinos. The Transmitter part is kept on palm and the recipient part on robot vehicle that moves as showed by the hand improvement. There are five distinctive kinds of movement locations based on movement of hand. Those are stop, forward, backward, a left and right. The obstacles will be detected and tracked through camera using python/OpenCV based contour tracing algorithm.

B. BASIC CNN COMPONENTS

Convolutional neural network layer consists of three types of layers, namely convolutional layer, pooling layer, and fully connected layer.

C. Convolutional Layer

The aim of CNN is to learn feature representations of the inputs. As shown in the below image (Fig. 1), Convolutional layer has several feature maps and it is the first layer from which features are extracted. Each neuron of the same feature map is used to extract local characteristics of different positions in the former layer. In order to obtain a new feature, the input feature maps are first convolved with a learned kernel (mask) and then the results are passed into a nonlinear activation function. We will get different feature maps by applying different filter masks. The typical activation function is softmax, sigmoid, tanh and Relu.

D. Pooling Layer

Secondary feature extraction can be done within the pooling layer of a CNN. It essentially reduces the dimensions of the feature maps and increases the robustness of feature extraction. Usually placed between two Convolutional layers, the size of feature maps in the pooling layer is determined according to the moving step of the masks. Also referred to as stride of masks. The two major pooling operations are average pooling and max pooling. We can also extract the high-level characteristics of inputs by stacking several Convolutional layers and pooling layers.

E. Fully connected Layer

We flatten our matrix in vector form and feed it into the fully connected layer. In a fully connected layer, all the neurons in the previous layer are connected to every single neuron of the current layer. No spatial information is preserved in the fully connected layers. An output layer follows the last fully connected layer. After combining all the neurons, we can see the entire neural network. For classification tasks, softmax regression is commonly used because it generates a well- performed probability distribution of the outputs to classify as dog, cat, car, truck, etc.

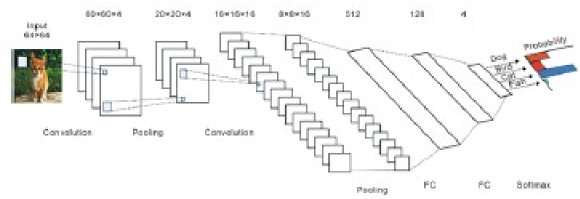

Fig. 1. CNN Architecture

F. Object Detection

We examined all the frameworks for key object detection characteristics like their speed and classification accuracy.

		Caffe	TensorFlow	Torch	Theano
Language		C++, Python	Python	Lua	Python
Pretrained		Yes ++	Yes (Inception)	Yes ++	Yes (Lasagne)
GPU		CUDA, Opecl	CUDA	CUDA	CUDA, Opecl
Good at RNN		No	Yes (Best)	Mediocre	Yes

Fig. 2. Classification of frameworks

Advantages of TensorFlow over other frameworks:

1) Easy deployment (Python pip package manager deployed by TensorFlow facilitates easy installation).
2) Better support for GPUs as compared to other models.
3) It provides high level APIs for building models.
4) It is extremely easy to do unconventional and hard-core changes.

Considering all the above parameters and the requirements of our project, we used the TensorFlow framework to identify object.

G. Pattern Detection

We unify the separate components of object detection into a single neural network. Our network uses features from the entire image to predict each bounding box. It also predicts all bounding boxes across all classes for an image simultaneously. This means our network reasons globally about the full image and all the objects in the image. The YOLO design enables end-to-end training and realtime speeds while maintaining high average precision.

Our system divides the input image into an S × S grid. If the center of an object falls into a grid cell, that grid cell is responsible for detecting that object.

Each grid cell predicts B bounding boxes and confidence scores for those boxes. 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. If no object exists in that cell, the confidence scores should be zero.

So that we get a class-specific confidence scores for each box and gives us a prediction about the object and identify according to the prediction.

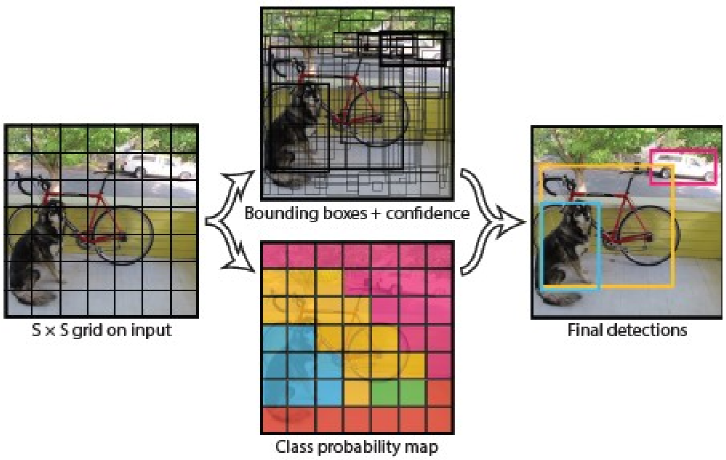

Fig. 3. Patter-Detection and Identify

III. Requirements

For making our project we used some software and hardware to make it more efficient.

A. Hardware

1) ESP32 CAM Module
The ESP32 Based Camera Module was developed by AI-Thinker. The controller is based on a 32-bit CPU has a combined Wi-Fi + Bluetooth/BLE Chip. It has a built-in 520 KB SRAM with an external 4M PSRAM. Its GPIO Pins have support like UART, SPI, I2C, PWM, ADC, and DAC. The module combines with the OV2640 Camera Module which has the highest Camera Resolution up to 1600 × 1200. The camera connects to the ESP32 CAM Board using a 24 pins gold plated connector. The board supports an SD Card of up to 4GB. The SD Card stores capture images.

2) FTDI Module
The FTDI(Future Technology Development International) adapter module is a complete package in which the FTDI chip is integrated with connectors, voltage regulators, Tx/Rx, and other breakout points. The module

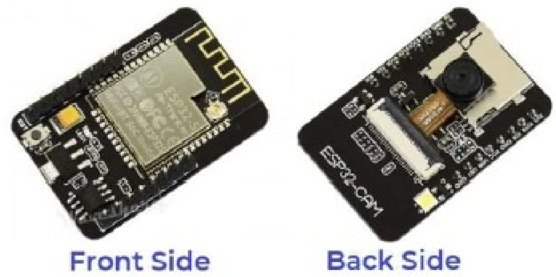

Fig. 4. ESP32 CAM Module

thus falls under the category of UART board and is mostly used for TTL serial communication.

3) Arduino
In a nutshell, an Arduino is an open hardware development board that can be used by tinkerers, hobbyists, and makers to design and build devices that interact with the real world.

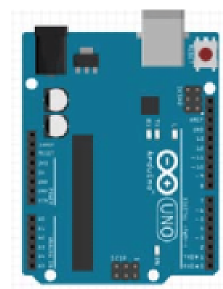

Fig. 5. Arduino Board

4) HC06
The HC-06 bluetooth module is a slave bluetooth module designed for wireless serial communication. It is a slave module meaning that it can receive serial data when serial data is sent out from a master bluetooth device(device able to send serial data through the air: smart phones, PC).

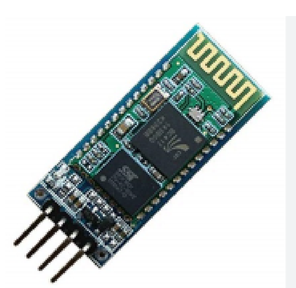

Fig. 6. Arduino Board

B. Software

1) Arduino IDE:
The Arduino Integrated Development Environment for writing code, a message area, a text console, a toolbar

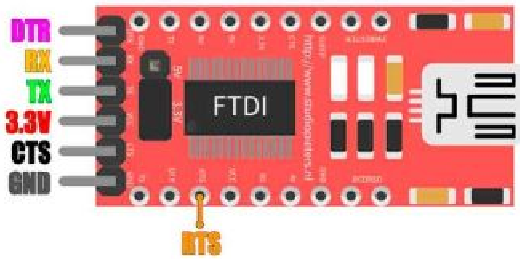

Fig. 7. FTDI Module

with buttons for common functions and a series of menus.

2) Visual Studio Code:
Visual Studio Code is a code editor for development operations like debugging, task running, and version control.

IV. Result and Discussion

A. Circuit Diagram

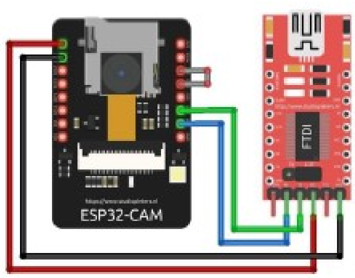

Fig. 8. Connection between ESP32 CAM Module and FTDI Module

As shown in figure 8 Connect the 5V and GND Pin of ESP32 to the 5V GND of the FTDI Module. Similarly, connect the Rx to UOT and Tx to UOR Pin. And the most important thing, you need to short the IO0 and GND Pin together. This is to put the device in programming mode. Once programming is done one can remove it.

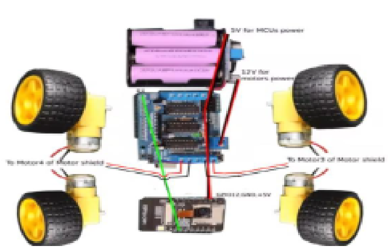

Fig. 9. Connection between all component

This is the total circuit diagram of the Project.

B. TensorFlow Workflow

The model is provided with two kinds of data for training the model. The image data is fed into the model, which learns

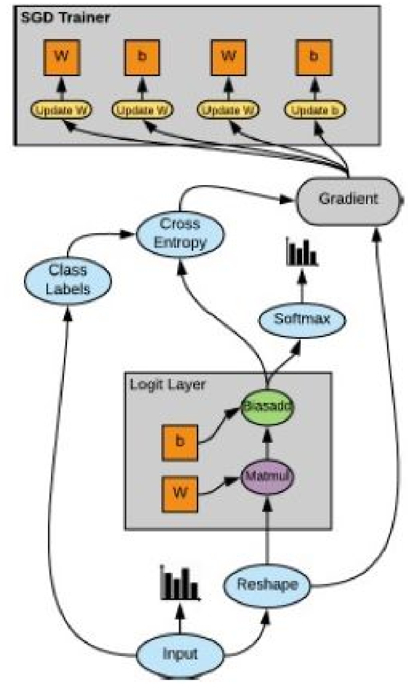

Fig. 10. TensorFlow Workflow

and predicts the output accordingly. The other kind of data is label data, which is provided at the end of the model to be compared with the predicted output.

1) Convolution with 64 different filters in size of 3 x 3.
2) Max pooling by 2 (Batch Normalization)
3) Convolution with 128 different filters in size of 3 x 3.
4) Max pooling by 2.
5) Convolution with 256 different filters in size 3 x 3.
6) Max pooling by 2.
7) Flatten the 3D output of the convolving operations.
8) Fully connected layer with 128 units
   - Dropout
   - Batch Normalization
9) Fully connected with 256 units
   - Dropout
   - Batch Normalization
10) Fully connected layers of 10 units.

V. Conclusion

As a result, we may infer that our system is capable of achieving all of the aforementioned goals and of overcoming the existing system's challenges. With our proposed system, surveillance is vastly improved. The designed system enabled us to achieve the following goals: real-time monitoring, reduced human intervention, and use of active sensors in the field.

The computational cost and time in a neural network is higher as compared to any other network models (R-CNN,

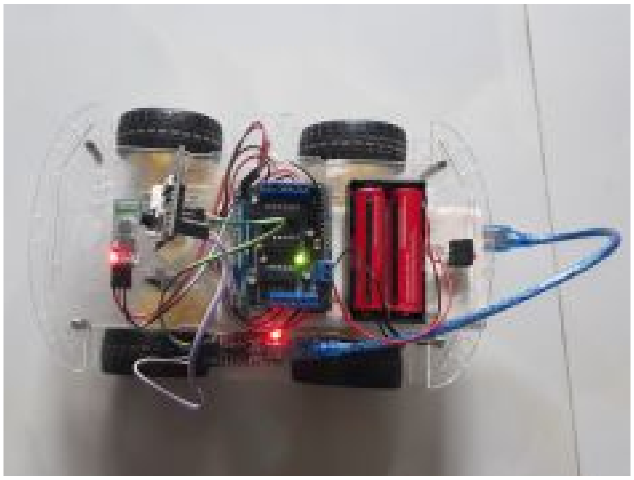

Fig. 11. Overview of Surveillance robot

Fig. 12. Object detection

Boltzmann machines, etc.) The most crucial requirement necessary to train CNN is a GPU (graphical processing unit). If the desktop/laptop used for training does not contain a GPU, the processing required for training a model increases which affects the performance. Therefore, it is imperative that the computer we use for training must have a GPU. The more, the model is trained the accurate it is. Hence, a huge amount of training data is required. This sometimes leads to the slow processing speed of the computer. These are the most of the challenges for us.

And in future we will work in different sectors also. Not only in surveillance sector, but also with different data set we can use it in different sector. Like, Traffic Sector, Agriculture system etc. And currently we are working with our countries perspective, in future we'll work for global.And also we have a plan for add the system of identifying the disease and the crop condition for more impactful result.

VI. Acknowledgment

We are overhelmed in all humbleness and gratefulness to acknowledge our depth to all those who have helped us to put these ideas, well above the level of simplicity and into something concrete. We would like to express our special thanks of gratitude to our respected teacher Md.Habibur Rahman, Lecturer, Department of Information and Communication Technology, Bangabandhu Sheikh Mujibur Rahman Digital University, Bangladesh for giving us the golden opportunity to do this project. By which we are able to build our project Monitoring Agri field using Machine Learning interfacing with IoT.

Any attempt at any level can't be satisfactorily completed without the support and guidance of our parents and friends. So, we also want to extend our sincere and heartfelt thanks to our parents and friends for their cordial support, valuable suggestion, and thoughts about our project.

References

[1] K. Marapalli, A. Bansode, P. Dundgekar and N. Rathod, "AIGER An Intelligent Vehicle for Military Purpose," 2021 7th International Conference on Advanced Computing and Communication Systems (ICACCS), Coimbatore, India, 2021, pp. 1052-1057, doi: 10.1109/ICACCS51430.2021.9441998.

[2] Ramalingam, B.; Mohan, R.E.; Pookkuttath, S.; Gómez, B.F.; Sairam Borusu, C.S.C.; Wee Teng, T.; Tamilselvam, Y.K. Remote Insects Trap Monitoring System Using Deep Learning Framework and IoT. Sensors 2020, 20, 5280. https://doi.org/10.3390/s20185280

[3] N. Materne and M. Inoue, "IoT Monitoring System for Early Detection of Agricultural Pests and Diseases," 2018 12th South East Asian Technical University Consortium (SEATUC), 2018, pp. 1-5, doi: 10.1109/SEATUC.2018.8788860.

[4] U. Shafi et al., "A Multi-Modal Approach for Crop Health Mapping Using Low Altitude Remote Sensing, Internet of Things (IoT) and Machine Learning," in IEEE Access, vol. 8, pp. 112708-112724, 2020, doi: 10.1109/ACCESS.2020.3002948.

[5] K. N. Bhanu, H. J. Jasmine and H. S. Mahadevaswamy, "Machine learning Implementation in IoT based Intelligent System for Agriculture," 2020 International Conference for Emerging Technology (INCET), 2020, pp. 1-5, doi: 10.1109/INCET49848.2020.9153978.

[6] M. L. N, A. E. Rao and M. P. Kalyan, "Real-Time Object Detection with Tensorflow Model Using Edge Computing Architecture," 2022 8th International Conference on Smart Structures and Systems (ICSSS), 2022, pp. 01-04, doi: 10.1109/ICSSS54381.2022.9782169.

[7] E. Bayhan, Z. Ozkan, M. Namdar and A. Basgumus, "Deep Learning Based Object Detection and Recognition of Unmanned Aerial Vehicles," 2021 3rd International Congress on Human-Computer Interaction, Optimization and Robotic Applications (HORA), 2021, pp. 1-5, doi: 10.1109/HORA52670.2021.9461279.

[8] D. K. Singh and R. Sobti, "Role of Internet of Things and Machine Learning in Precision Agriculture: A Short Review," 2021 6th International Conference on Signal Processing, Computing and Control (ISPCC), 2021, pp. 750-754, doi: 10.1109/ISPCC53510.2021.9609427.

[9] K. Marapalli, A. Bansode, P. Dundgekar and N. Rathod, "AIGER An Intelligent Vehicle for Military Purpose," 2021 7th International Conference on Advanced Computing and Communication Systems (ICACCS), Coimbatore, India, 2021, pp. 1052-1057, doi: 10.1109/ICACCS51430.2021.9441998.

[10] C. Tang, Y. Feng, X. Yang, C. Zheng and Y. Zhou, "The Object Detection Based on Deep Learning," 2017 4th International Conference on Information Science and Control Engineering (ICISCE), Changsha, China, 2017, pp. 723-728, doi: 10.1109/ICISCE.2017.156.

[11] Z. Shi, "Object Detection Models and Research Directions," 2021 IEEE International Conference on Consumer Electronics and Computer Engineering (ICCECE), Guangzhou, China, 2021, pp. 546-550, doi: 10.1109/ICCECE51280.2021.9342049.

[12] K. Li and L. Cao, "A Review of Object Detection Techniques," 2020 5th International Conference on Electromechanical Control Technology and Transportation (ICECTT), Nanchang, China, 2020, pp. 385-390, doi: 10.1109/ICECTT50890.2020.00091.

[13] W. Tarimo, M. M.Sabra and S. Hendre, "Real-Time Deep Learning-Based Object Detection Framework," 2020 IEEE Symposium Series on Computational Intelligence (SSCI), Canberra, ACT, Australia, 2020, pp. 1829-1836, doi: 10.1109/SSCI47803.2020.9308493.

[14] Z. -Q. Zhao, P. Zheng, S. -T. Xu and X. Wu, "Object Detection With Deep Learning: A Review," in IEEE Transactions on Neural Networks and Learning Systems, vol. 30, no. 11, pp. 3212-3232, Nov. 2019, doi: 10.1109/TNNLS.2018.2876865.