

Chapter 1: Introduction

1.1 Motivation:

Unmanned Aerial Vehicles (UAVs) have received attention in the last decade because of their low cost, small size, and programmable features. Indoor autonomous navigation of UAVs has been actively studied in robotics community. Indoor applications have less boundary condition compared to outdoor applications. The main purpose is to make the inventory process completely autonomous. Indoor navigation of an UAV is used in different field such as in warehouse operations, where it can be use in inventory management, intra-logistics and inspection and surveillance etc.

In manufacturing and production companies, various tasks are gradually being automated – to be performed by robots instead of human labours. These automation robots include Unmanned Ground Vehicle (UGV) and Unmanned Aerial Vehicle (UAV). The tendency of using drone technologies in the industry completely changes the business model and redraws industrial landscapes. This trend affects a huge range of industry areas, such as infrastructure, transport, insurance, media, entertainment, telecommunication, agriculture, mining, and security. Drone-based solutions are the most appropriate for sectors interested in two principal aspects of mobility and high data quality. One of the most significant aspects of the development of autonomous systems based on UAV is their take-off/landing system. It should work autonomously, provide the specified accuracy, while the equipment installed on the UAV must weigh as little as possible and consume as little computing power as possible.

The motivation of this project is to detect the object robustly in indoor environment with good accuracy by using the images obtained from drone camera with the help of computer vision algorithm.

1.1 Objective

The objective of the thesis is:

- Detect predetermined object in video streams using computer vision approach.
- Autonomously navigate the UAV in a given relative position

1.2 Performance factors and definitions

Some performance factors had considered while implementing this model.

Power consumption must be the important factor in UAVs applications. If the system will become portable, flexible then power must be kept low. Here, the StromPi battery unit gives portable supply to on board UAV computer.

Robustness is getting from the minimum errors and less disturbance of the moving objects.

Scalability is basically depends on the how large is the working area. This project is developed in indoor environment as it requires large area to cover the required task and detect the multiple objects.

Complexity is related to the project design i.e., particular system. It should be easy to install in new environment. It must not take too much time to calibrate some technical things and so on.

Portability describes how easy to move this hardware from one place to another. In short, it depends on weight, size and power module.

Accuracy is an important parameter while detecting the objects in indoor environment. Because of light conditions, the objects might get some time to detect by UAV.

Latency is important when considering a moving target, since a high latency will result in a high inaccuracy. Latency can be seen as the time in seconds from the system acquires data until it is finished processing it

1.3 Limitations

1.4 Thesis organization

The structure of this thesis is as follows.

- Chapter 1 gives the introduction and background of thesis topic. It's become easy to understand the topic objective and basic information of the project topic from chapter 1.
- Chapter 2 reviews the indoor navigation systems used by UAV. This chapter helps to find the ongoing and previous research has been done in this related indoor navigation field. The second chapter has a way to get some solutions to implement this thesis topic.
- Chapter 3 explain the how first objective of the thesis has achieved using deep learning method. In that, different object detection techniques have been explained.
- Chapter 4 gives information about the overall system architecture of UAV. Here, all the hardware related work and used components has used in the UAV is explained. This chapter presents the general overview of hardware components and software design.
- Chapter 5

II

Literature Survey

This chapter will review the theory and survey related to indoor positioning of UAV. This section will introduce a several methods to obtain a robust autonomous navigation [1], autonomous landing on unmanned surface vehicle [2], and various applications of UAV used in indoor environment [3].

2.1 Review of state-of-the-art Indoor UAV Platforms

From past few years, there has been number of innovations are done in warehouses for indoor navigation system. Although it's a challenging task and following research has helped to choose the appropriate method require to complete this thesis topic.

2.1.1 Autonomous warehouse inventory management [4] research explains the working of unmanned aerial vehicle (UAV) and unmanned ground vehicle (UGV). The UGV is basically a ground reference to get the position for UAV. This approach is somehow similar to this thesis topic. They have presented a novel technique for the automation of warehouse task. They used UGV as a carrying platform in GPS denied environment while UAV is used as the mobile scanner to scan the barcode on goods. The UAV has mounted a camera on front side which will scan the barcode. The UGV starts to navigate the rows of racks which carrying the UAV. In the given rack, there are coded markers has pasted which indicates the ID of appropriate rack number as shown ion fig. 1.

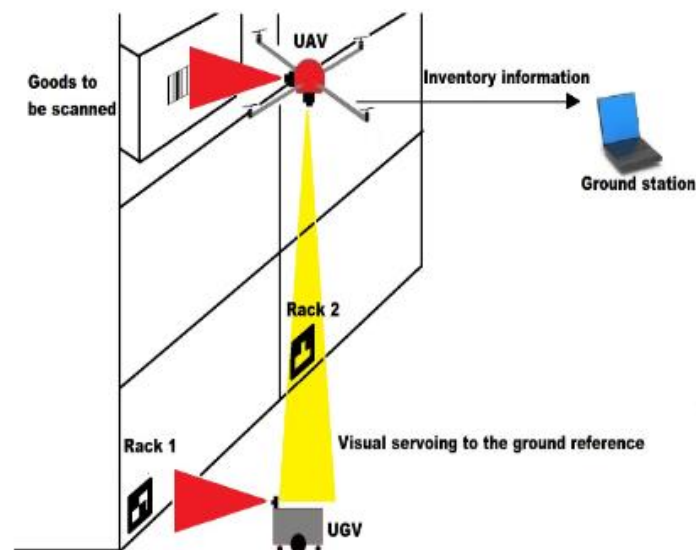


Fig. 1 [4]

The UGV is also having the front facing camera to detect the marker. When UGV detect the marker, it will stop at that moment and UAV will start to fly vertically for scanning the barcodes placed on goods. When UAV reaches at top, it UGV change its position and moves to next rack marker. As UGV is ground reference for UAV, it will follow UGV. Now, the UAV is located on top of second rank. It will start to scan the barcode from top to bottom. The process will go until the whole row is completely scanned.

The UAV also has a down facing camera to detect the marker which is placed on upper side of UGV for vision based target tracking. When the whole row of racks has scanned successfully, the UAV will land on UGV from the Augmented Reality (AR) marker and it will recharge its batteries before moving forward to next row.

2.1.2 The author from [1] presented **the robust navigation of UAVs in warehouses** using another method from previous research. They have developed low-cost sensing system using Extended Kalman Filter (EKF). In their research paper, they have mainly focused on the three things: 1) neglect the outliers, inherent drift using Mahalanobis norms, 2) incorporation of visual SLAM by introducing pseudo-covariance, 3) recognition of floor lanes to get position and yaw measurement.

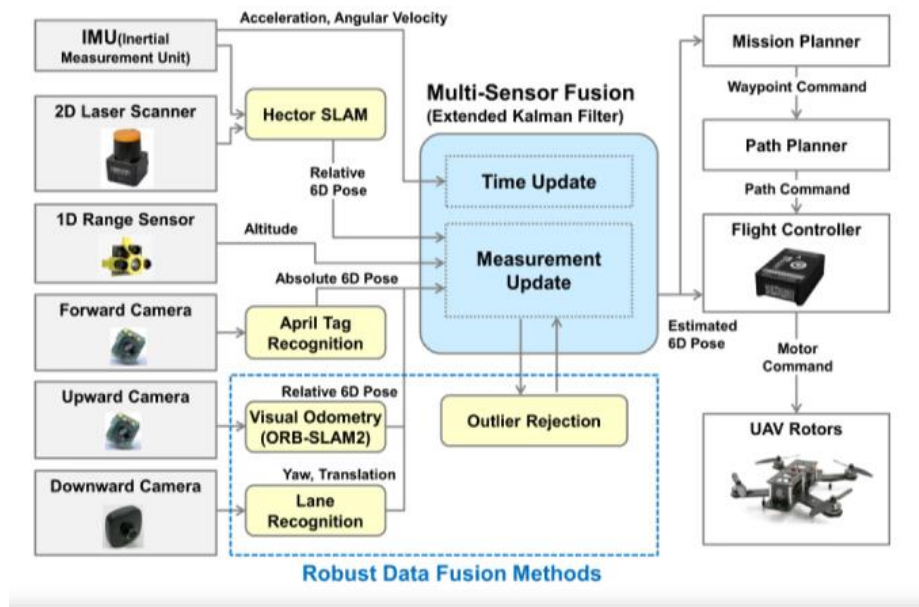


Fig. 2 [1]

In this research paper, author suggests multi-sensor fusion method with the estimated pose and applies Extended Kalman filter (EKF) using three cameras, 2D laser scanner, 1D range sensor, and IMU as shown in fig. 2. EKF will acquire the data from sensors in the form of relative and absolute angles, poses and distance. The SLAM method provides the 6D relative positions i.e. x, y, z, roll, pitch and yaw of the UAV. While absolute positions will get from April Tag marker detection algorithm.

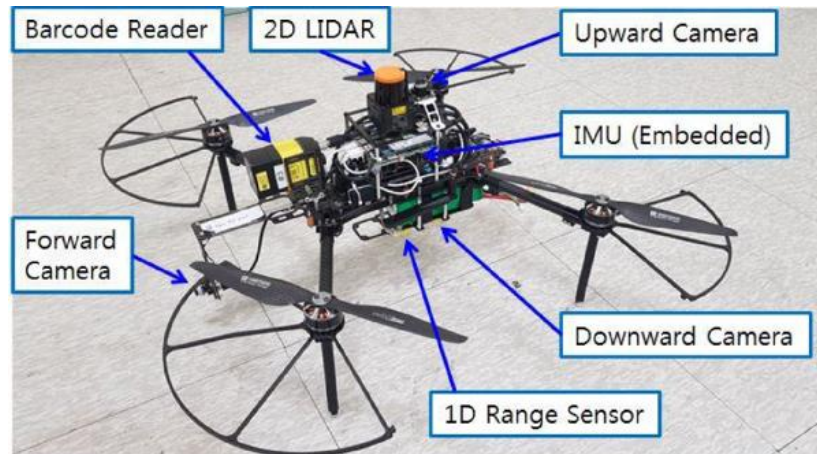


Fig. 3 [1]

The 1D range sensor is used as an altitude measurement to measure the distance up to 14 meters from floor to UAV. The forward camera will detect the April Tag maker to get the absolute 6D pose data of UAV. The EKF will sends the estimated UAV poses to flight controller and ground station as shown in Fig.2.

2.1.3 The research paper [5] explained about the UAV landing on mobile collaborative ground robot reference on the basis of IR marker detection. They have used two robots i.e., unmanned aerial vehicles (UAV) and unmanned ground vehicles (UGV) for precise localization. Using the help of 2D LIDAR sensors, camera and ultrasonic system they will get the precise localization.



Fig. 4 [5]

They have placed camera on mobile robot and IR markers are on the UAV. The localization is possible with fusing two IR markers patterns i.e., small and big marker. The small marker will help to take-off and land the UAV on robot and the bigger will use to calculate the height above 1 m. The ultrasonic sensors are connected to on board computer of the UGV to solve the problem of precise position estimation on high altitudes.

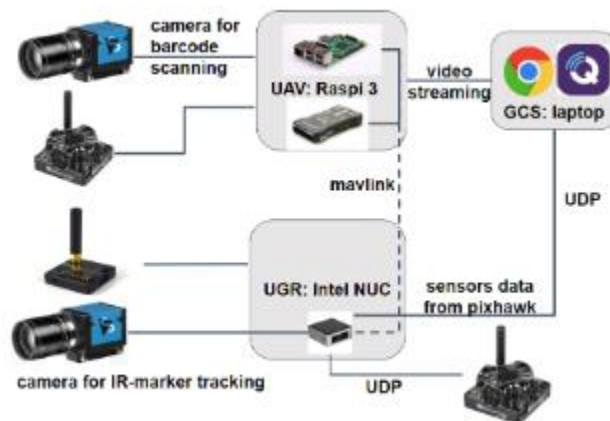


Fig. 5 [5]

The above figure is the communication setup of UAV and UGV where camera to be used for scanning the barcode is connected to UAV on board computer i.e., raspberry pi. The UAV can calculate the global coordinates using its coordinate relative to UGV.

2.1.4 The authors from [6] presented real-time implementation of object detector and tracking system for AR Drone 2 using SSD neural network. Their aim is to detect and track the target object using drone camera. Using the front camera, they have developed the tracking algorithm which calculate the parameters such as roll, pitch, yaw and altitude. However, these parameters are controlled by PID controller which takes the input as a position and distance of the target object.

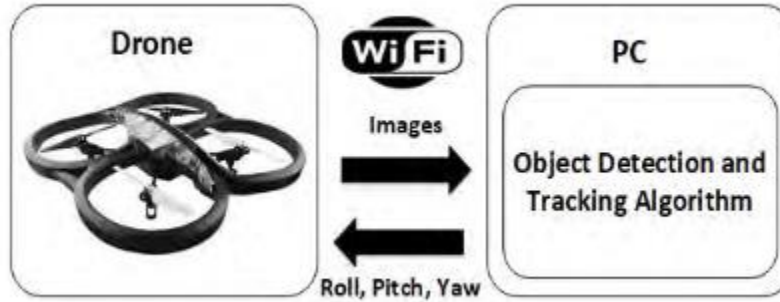


Fig. 6 [6]

For detecting the object, drone sends the captured image to PC via Wi-Fi using front camera. The PC will train the images in CNN which used single shot detector (SSD) architecture. When the object will detect in image, then it returns the bounding box of detected object. Using that bounding box, it will find the center of the detected object. This center will use as an input to PID controllers. This PID controller parameter roll will calculate the position of the detected object. The benefit of using the SSD architecture is that object will detected with high accuracy by reducing the computational time.

Chapter 3

Computer vision preliminaries

3.1 Overview

Computer vision is a field of artificial intelligence that trains computers to interpret and understand the visual world. Using digital images from cameras and videos and deep learning models, machines can accurately identify and classify objects and then detect the appropriate identified objects. Computer vision has some abilities to make it used in AI field such as, it provides the computing power has become more affordable and easily accessible and new algorithms like convolutional neural networks can take advantage of the hardware and software capabilities. These effects are playing more vital role in today's world. In last decade, the computer vision technique is more accurate than humans to detect the visual inputs as the accuracy is increased from 50% to 99% for detection and classifications of the images.

3.2 Deep learning and Neural Networks:

Over the last years deep learning methods have been shown to outperform previous state-of-the-art machine learning techniques in several fields, with computer vision being one of the most prominent cases. Deep learning allows computational models of multiple processing layers to learn and represent data with multiple levels of abstraction mimicking how the brain perceives and understands multimodal information, thus implicitly capturing intricate structures of large-scale data. Deep learning is a rich family of methods, encompassing neural networks, hierarchical probabilistic models, and a variety of unsupervised and supervised feature learning algorithms.

The recent surge of interest in deep learning methods is due to the fact that they have been shown to outperform previous state-of-the-art techniques in several tasks, as well as the abundance of complex data from different sources (e.g., visual, audio, medical, social, and sensor).

Deep learning (DL) in Artificial Intelligence (AI) has recently gained a significant interest. It is used in a wide range of applications such as autonomous systems, facial recognition, self-driving cars, image and speech recognition, classification, and object detection. Among the most promising systems that can utilize Deep Learning are Unmanned Aerial Vehicles (UAVs), which are becoming an attractive solution for a wide range of applications.

Recently, Convolutional Neural Networks (CNNs) have achieved great results indifferent fields of recognition, detection, and classification, especially in computer vision. CNN is a class of deep neural networks, and is mostly applied to analyze visual imagery. For the application of object detection and classification, CNN is considered as a very powerful tool. CNNs are bio-logically inspired hierarchical models that can be trained to perform a variety of detection, recognition, and segmentation.

Object detection can be grouped into two type's namely specific objects and generic objects. There is an issue with matching problem in specific objects (Fig.1(a)). In generic object type, we have to detect instances of some predefined objects (Fig.1(b)).

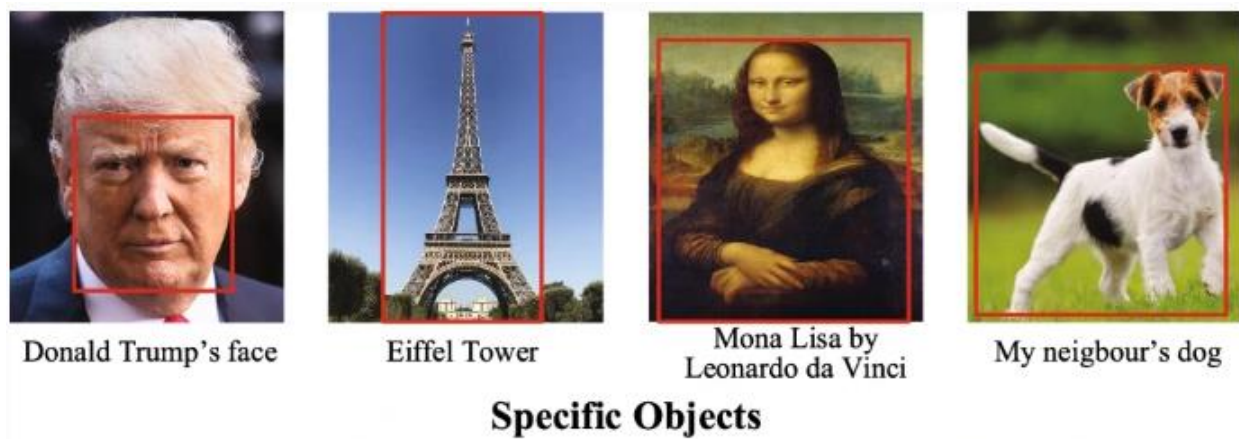


Figure 3.1

Deep learning has altered a wide range of machine learning tasks, from image classification and video processing to speech recognition and natural language understanding.

Convolutional Neural Networks (CNNs), the most representative models of deep learning, are able to exploit the basic properties underlying natural signals: translation invariance, local connectivity, and compositional hierarchies.

A typical CNN has hierarchical structure and composed of numbers of layers that shown in below figure.

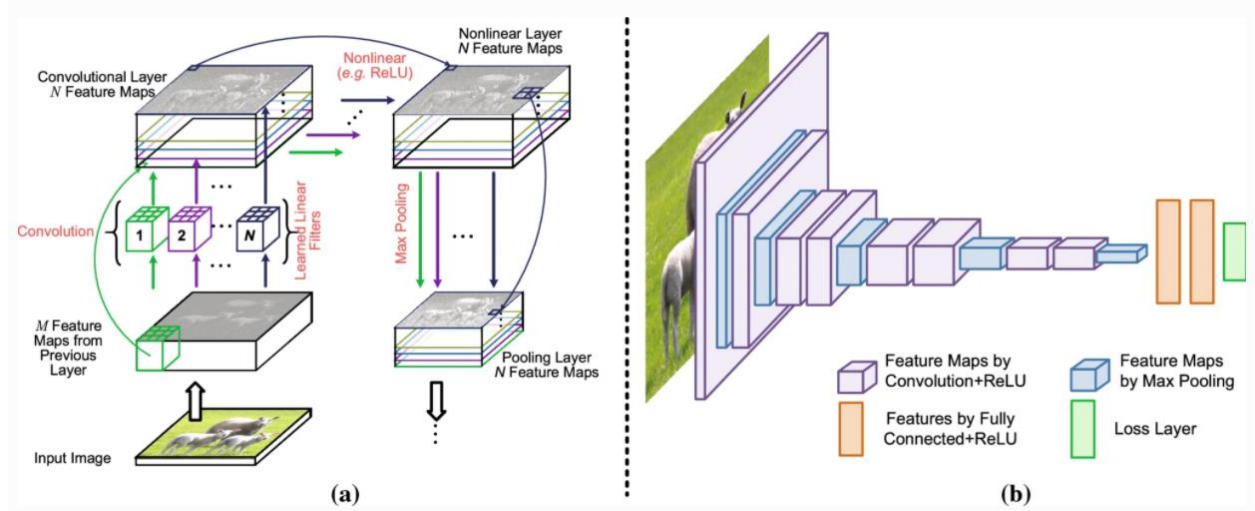


Figure 3.2

We have a convolution

$$\mathbf{x}^{l-1} * \mathbf{w}^l$$

between an input feature map \mathbf{x}^{l-1} at a feature map from previous layer $l-1$, convolved with a 2D convolutional kernel (or filter or weights) \mathbf{w}^l . This convolution appears over a sequence of layers, subject to a nonlinear operation σ , such that

$$\mathbf{x}_j^l = \sigma \left(\sum_{i=1}^{N^{l-1}} \mathbf{x}_i^{l-1} * \mathbf{w}_{i,j}^l + b_j^l \right)$$

with a convolution now between the N^{l-1} input feature maps \mathbf{x}_i^{l-1} and the corresponding kernel $\mathbf{w}_{i,j}^l$, plus a bias term b_j^l . The elementwise nonlinear function $\sigma(\cdot)$ is typically a rectified linear unit (ReLU) for each element,

$$\sigma(x) = \max\{x, 0\}.$$

Finally, pooling corresponds to the downsampling/upsampling of feature maps. These three operations (convolution, nonlinearity, pooling) are illustrated in Fig. 8a; CNNs having a large number of layers, a “deep” network, are referred to as Deep CNNs (DCNNs), with a typical DCNN architecture illustrated in Fig. 3(b). Most layers of a CNN consist of a number of feature maps, within which each pixel acts like a neuron.

3.3 Datasets

Datasets have played a key role throughout the history of object recognition research, not only as a common ground for measuring and comparing the performance of competing algorithms, but also pushing the field towards increasingly complex and challenging problems.

Recently, deep learning techniques have brought tremendous success to many visual recognition problems, and it is the large amounts of annotated data which play a key role in their success. Access to large numbers of images on the Internet makes it possible to build comprehensive datasets in order to capture a vast richness and diversity of objects, enabling unprecedented performance in object recognition.



Figure 3.3

Fig. 3.3 Images with object annotations from PASCAL VOC, ILSVRC, MS COCO and Open images.

For object detection there are mainly four famous datasets: PASCAL VOC, ImageNet, MS COCO and Open Images.

PASCAL VOC is a multi-year effort devoted to the creation and maintenance of a series of benchmark datasets for classification and object detection, creating the precedent for standardized evaluation of recognition algorithms in the form of annual competitions.

ILSVRC, the ImageNet Large Scale Visual Recognition Challenge is derived from ImageNet scaling up PASCAL VOC's goal of standardized training and evaluation of detection algorithms by more than an order of magnitude in the number of object classes and images. ImageNet1000, a subset of ImageNet images with 1000 different object categories and a total of 1.2 million images, has been fixed to provide a standardized benchmark for the ILSVRC image classification challenge.

MS COCO is a response to the criticism of ImageNet that objects in its dataset tend to be large and well centered, making the ImageNet dataset atypical of real-world scenarios. To push for richer image understanding, researchers created the MS COCO database containing complex everyday scenes with common objects in their natural context, closer to real life, where objects are labeled using fully-segmented instances to provide more accurate detector evaluation.

OICOD (the Open Image Challenge Object Detection) is derived from Open Images V4 (now V5 in 2019) currently the largest publicly available object

detection dataset. OICOD is different from previous large scale object detection datasets like ILSVRC and MS COCO, not merely in terms of the significantly increased number of classes, images, bounding box annotations and instance segmentation mask annotations, but also regarding the annotation process. In ILSVRC and MS COCO, instances of all classes in the dataset are exhaustively annotated, whereas for Open Images V4 a classifier was applied to each image and only those labels with sufficiently high scores were sent for human verification. Therefore in OICOD only the object instances of human-confirmed positive labels are annotated.

3.4 Detection Frameworks:

As mentioned earlier in introduction part that there are several object detection frameworks are present. In this chapter, the different CNNs are explained.

3.4.1 Region Based Computational Neural Network (R-CNN):

Rather than operating on a large number of regions, the RCNN algorithm proposes a set of boxes in the image and checks to see if any of them contain any objects. To extract these boxes from an image, RCNN employs selective search (these boxes are called regions).

First, let's define selective search and how it distinguishes between different regions. An object is made up of four regions: different scales, colors, textures, and enclosure. Selective search recognizes these trends in the picture and suggests different regions based on them. Here's how targeted quest works in a nutshell:

Here, Image should be the first input:



Figure 3.4 (A)

Then, it generates initial sub-segmentations so that we have multiple regions from this image:



Figure 3.4 (B)

The technique then combines the similar regions to form a larger region (based on color similarity, texture similarity, size similarity, and shape compatibility):



Figure 3.4 (C)

- Finally, these regions then produce the final object locations (Region of Interest).

The following is a quick rundown of the steps taken by RCNN to detect objects:

1. We start with a convolutional neural network that has already been educated.
2. The model is then retrained.
The number of classes that need to be detected is used to train the network's last layer.
3. Get the Region of Interest for each picture as the third stage.
After that, we reshape all of these regions to fit the CNN input size.
4. We train SVM to identify objects and history after we get the regions.
One binary SVM is trained for each class.
5. Finally, for each defined entity in the picture, we train a linear regression model to produce tighter bounding boxes.

Here is the input image,



Figure 3.5 (A)

Then, using some proposal method (for example, selective search as seen above), we get the Regions of Interest (ROI):

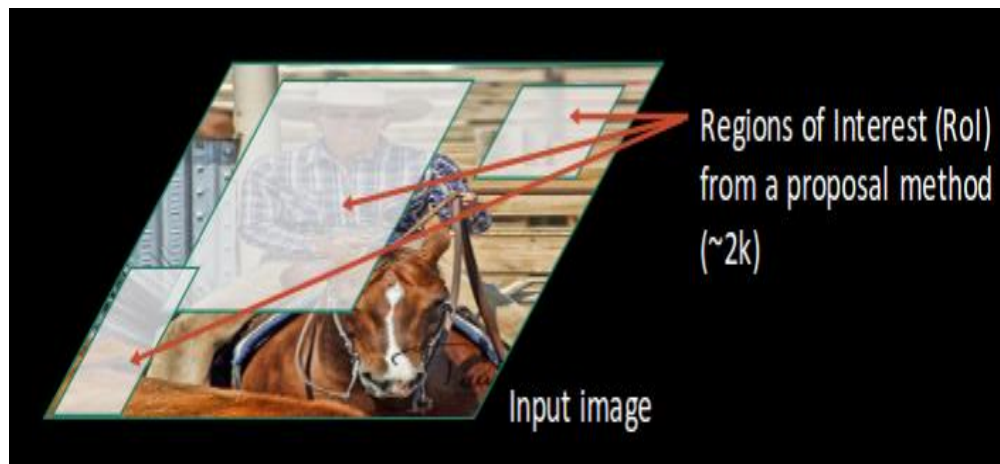


Figure 3.5 (B)

After that, all of these regions are reshaped according to the CNN's input, and each region is transferred to the ConvNet.

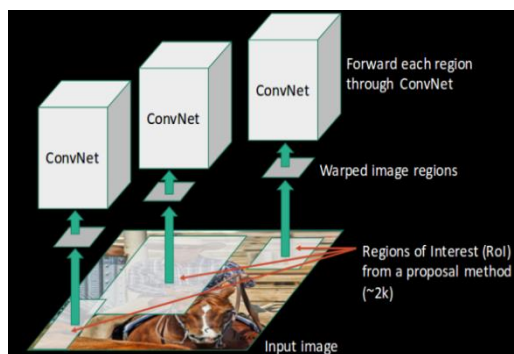


Figure 3.5 (C)

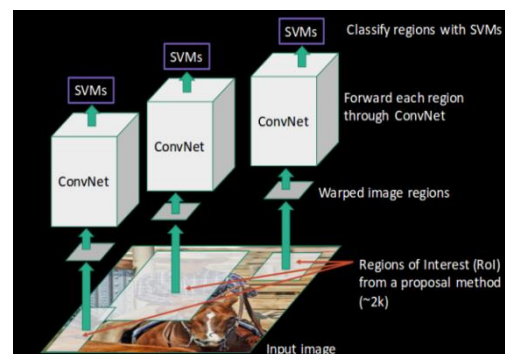


Figure 3.5 (D)

SVMs are used to segment these regions into separate groups after CNN extracts features for each region. Finally, for each specified field, a bounding box regression (Bbox reg) is used to predict the bounding boxes.

Problems with RCNN

We've seen how RCNN can help with object detection so far. However, this method has its own set of limitations. The following steps make training an RCNN model both costly and time-consuming:

- Using a selective scan, extracting 2,000 regions for each picture
- Use CNN to extract features for each picture field. The number of CNN features would be $N \times 2,000$ if we have N videos.

-

The entire object detection mechanism with RCNN is divided into three models:

1. Function extraction using CNN
2. Object recognition using a linear SVM classifier
3. Bounding box tightening using a regression model

3.4.2 Fast R-CNN

We feed the input image to the CNN in Quick RCNN, which then generates the convolutional feature maps. The regions of proposals are extracted using these maps. The proposed regions are then reshaped into a fixed size using a RoI pooling layer so that they can be fed into a completely linked network.

To make the idea clearer, let's break it down into steps:

1. As with the previous two techniques, we start with a picture.
2. The picture is fed into a ConvNet, which creates the Regions of Interest.
3. On all of these regions, a RoI pooling layer is added to reshape them according to the ConvNet's feedback. After that, each area is linked to a larger network.

4. To output groups, a softmax layer is used on top of the fully connected network. A linear regression layer is used in parallel with the softmax layer to produce bounding box coordinates for predicted groups.

Rather than using three separate models (as in RCNN), Quick RCNN employs a single model that extracts features from regions, divides them into classes, and At the same time, create boundary boxes for the defined groups.

Below is the visualization of each step that we discussed above.

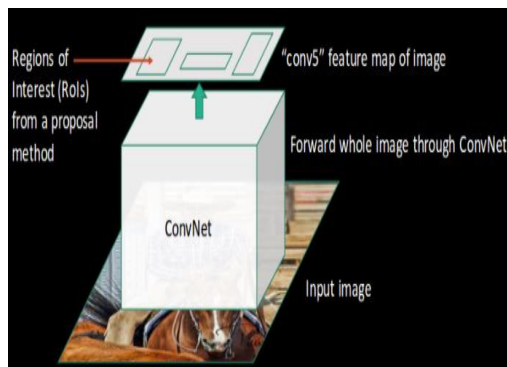


Figure 3.6 (A)

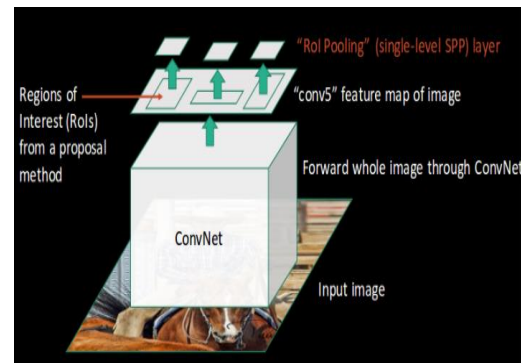


Figure 3.6 (B)

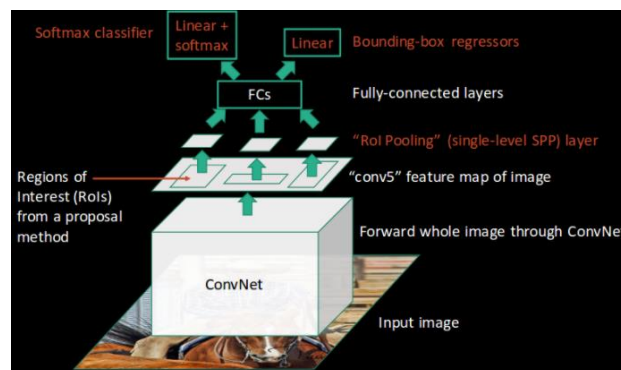


Figure 3.6 (C)

Problems with Fast RCNN

There are some issues with Quick RCNN.

It also proposes using selective search to find the Regions of Interest, which is a slow and time-consuming process.

Detecting objects takes around 2 seconds per image, which is much faster than RCNN.

However, when dealing with massive real-world datasets, even a Fast RCNN becomes slow.

3.4.3 Faster R-CNN:

Faster RCNN is a variant of Fast RCNN that has been tweaked. The main difference is that Fast RCNN generates Regions of Interest using selective search, whereas Faster RCNN uses RPN (Region Proposal Network). RPN takes image feature maps as input and outputs a collection of object proposals, each with a ranking for objectness.

The below steps are typically followed in a Faster RCNN approach:

1. We take an image and pass it to the ConvNet, which returns the image's feature map.
2. On these function maps, a region proposal network is implemented. The object proposals are returned, along with their objectness ranking.
3. These proposals are subjected to a RoI pooling layer, which reduces the size of all proposals to the same level.
4. To classify and output the bounding boxes for artifacts, the proposals are moved to a completely connected layer with a softmax layer and a linear regression layer at its top.

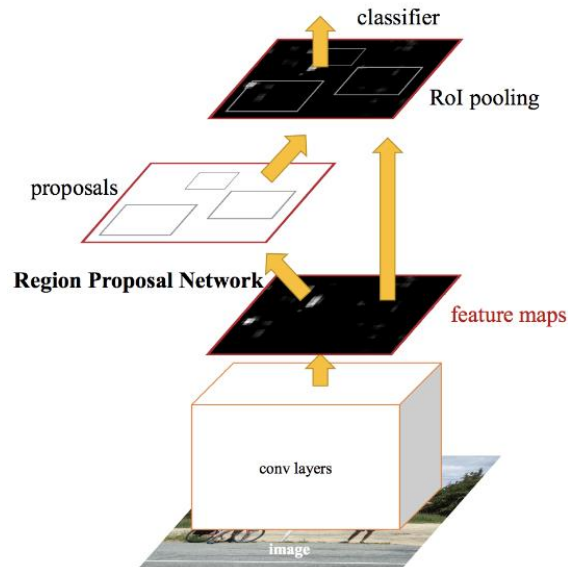


Figure 3.7 (A)

First, we'll talk about the Regional Proposal Network (RPN). RPN is the region in the image where an object might be located. We classified the objected area as foreground class, while the area where the object is not present is labeled as background class.

Anchor boxes are created by RPN using a sliding window. Anchor boxes are predefined bounding boxes that come in a variety of sizes and shapes.

Once we have generated anchor boxes then next step is to find out IoU (Intersection over union).



Figure 3.7 (B)

The overlapping region from the previous figure is referred to as IoU. If the overlapping region is greater than 50%, the object will be identified by the box. Foreground class refers to the anchor box with the highest IoU.

3.4.4 Single Shot Detector (SSD):

- Input image gives to VGG16 network to extract feature map.
- SSD makes 8732 predictions for every single class using 6 convolutional layers.
- This 6 CNN layers performed classification and detection task.

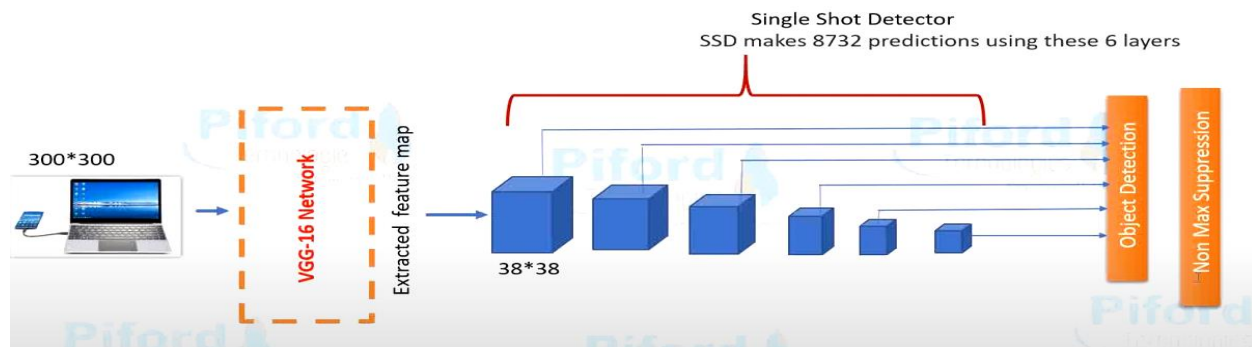


Figure 3.8 (A)

- We used non-max suppression just to remove duplicate predictions.

Let's take an example for better understanding of SSD.

Below image is the input of SSD and it must have ground truth box for each image as we can see in figure yellow and green box.



Figure 3.8 (B)

After this, we are having convolutional layers. The task of this CNN layers is to check boxes of different aspect ratios and each locations with different sizes.



Figure 3.8 (C)

As we discussed earlier, there are 8732 boxes per object in above figure. So for laptop there are 8732 bounding boxes and for mobile have also 8732 boxes. We can see that there are multiple boxes are overlapping each other. We used IoU here to find out highest overlapping box. Using IoU concept we can detect object easily.

3.4.5 You Only Look Once (YOLO):

YOLO predicts the bounding box coordinates and class probabilities for these boxes using the entire image as a single case. The most significant benefit of using YOLO is its incredible speed; it can process 45 frames per second. YOLO is also aware of the concept of abstract object representation.

Following steps will help to understand the YOLO algorithm easily.

Suppose following image is the input for YOLO.

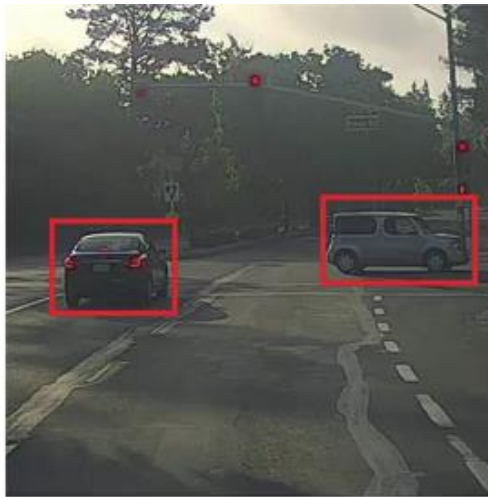


Figure 3.9 (A)

The grid for YOLO framework would be 3 x 3.

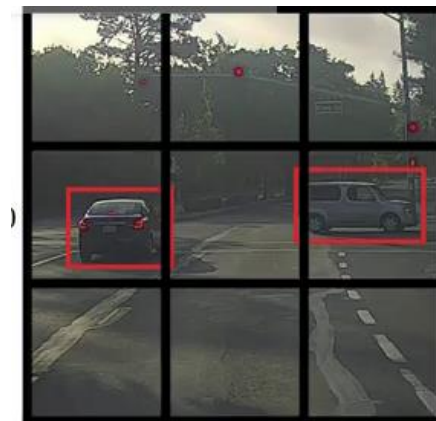


Figure 3.9 (B)

Each grid has an image localization and image classification. To comprehend the YOLO algorithm, one must first determine what is being expected. Finally, it can able to predict an object's class and the bounding box that defines its position.

Bounding box can be defined into five parts:

- i. p_c : It shows probability of whether object is present in the bounding box.
- ii. b_x, b_y : Defines the center of bounding box
- iii. b_h and b_w : height and width of bounding box
- iv. $c_1, c_2, c_3, \dots, c_n$: It represents the class of an object such as cars, bikes, pedestrian etc.

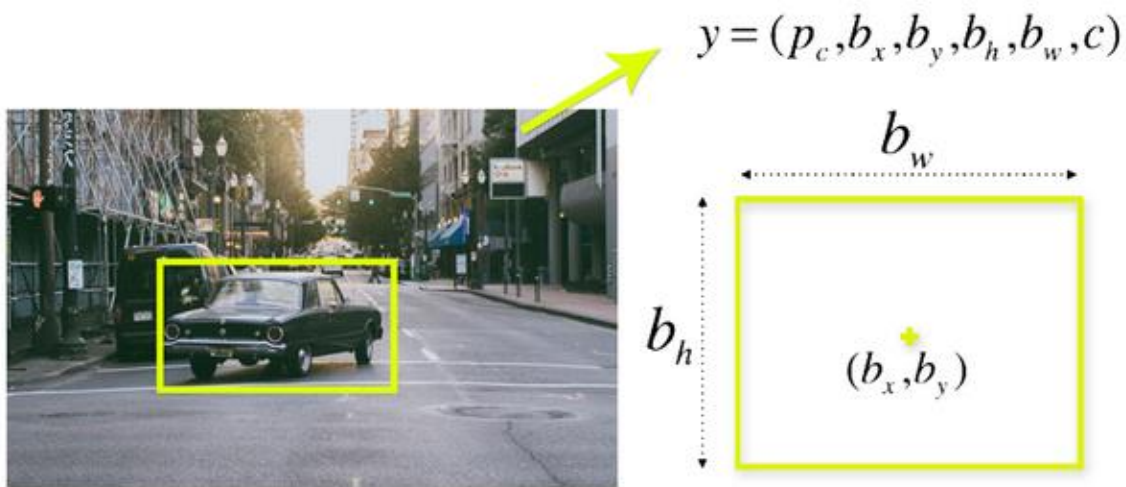
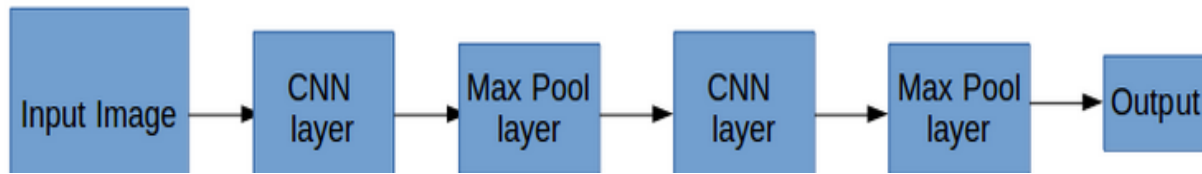


Figure 3.9 (C)

The algorithm to detect the object of the RCNN, Fast-RCNN and Faster-RCNN is on the basis region of interest of an image. And they defined these ROI using CNNs. But YOLO works differently. YOLO is not searching for region of interest. Instead, spill pictures into cells using a 19 x 19 grid. Each cell is in charge of forecasting five bounding boxes (in case there is more than one object in this cell). As a result, for one picture, it gets 1805 bounding boxes.



The majority of these cells and bounding boxes will be empty. As a result, we estimate the value p_c , which is used in a method known as **non-max suppression** to delete boxes with low object likelihood and bounding boxes with the highest shared space.

How non-max suppression works?

While detecting the object in real-time, the object detection algorithm detects one object in many times. The detected object has surrounded with multiple bounding boxes as shown in below figure. Every image has p_c value i.e. probability value. So, it selects the bounding box which has highest probability i.e. 0.9 in this case. Secondly, it takes highest IoU of the another box. Therefore, 0.6 and 0.7 probabilities are suppressed. This will goes until all the boxes after either selecting or compressing the bounding boxes.



Figure 3.5 (D)

Chapter 4

System Architecture Design For Unmanned Aerial Vehicle

4.1 Overview

As described in chapter 1, the purpose of this research is to investigate and develop a system for vision based navigation using UAV in the context of indoor environment.

Chapter 2 describes the literature review and also the materials and various theoretical methods applied in this research. Chapter 3 explains about the methods used in this research for predetermined object in video streams using computer vision approach.

This chapter presents the design of the system architecture for UAV. This section divided into two parts. First part covers the hardware design which included the UAV structure, serial connections and hardware setup. The second part gives the information about software which used to control the UAV.

4.2 Hardware Design

The system architecture for indoor navigation on board system consists of quadcopter UAV (UAV name), an autopilot (Pixhawk), a microprocessor Raspberry Pi 4, Raspberry Pi camera, Intel Real-Sense depth camera D435, 4500 mAh Lipo Pack battery, Radio Control (Tx/Rx), StromPi Power and Battery Module WiFi connection and GSC (Mission Planner software) with additional component. Figure shows the hardware component attached to the UAV.

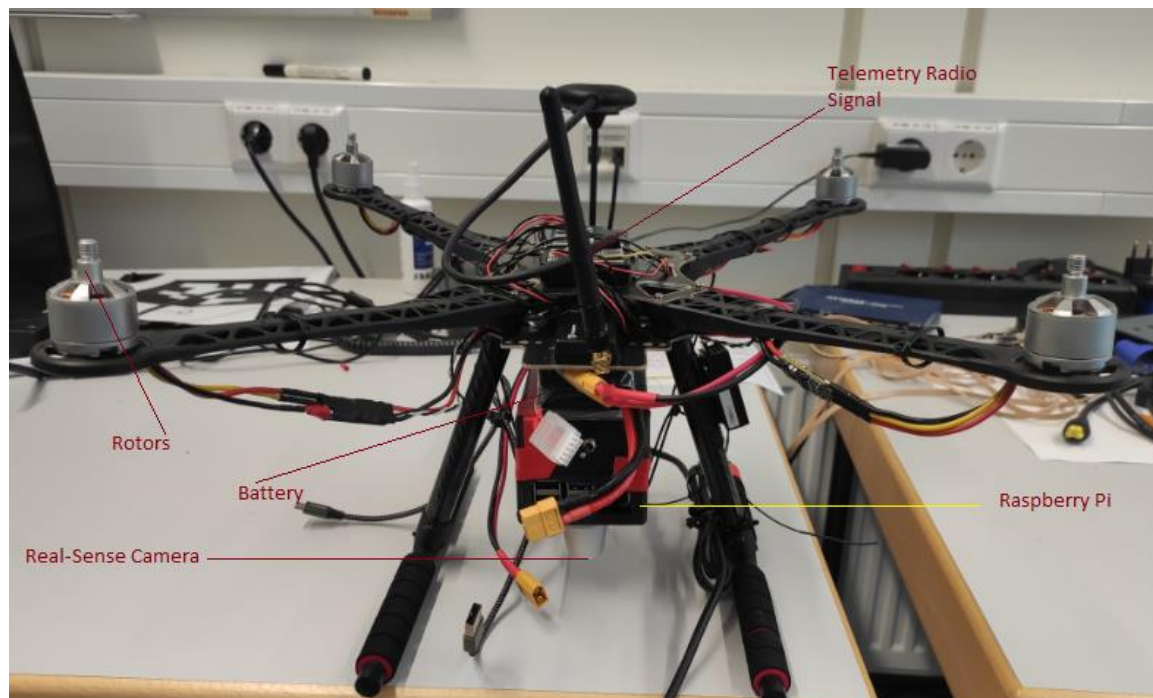


Figure 4.1

4.2.1 UAV Frame

4.2.2 Pixhawk Autopilot

Pixhawk 4 is a flight controller unit (FCU) open source autopilot designed and manufactured by Holybro and the PX4 team. There are different types of versions are available in Pixhawk. Pixhawk 4 has used for these project. It comes preinstalled with latest px4 firmware. Pixhawk FCU has in built in-built accelerometer, gyroscope, barometer and magnetometer.



Figure 4.2

These FCU has 17 Ports to connect UAV motors, GPS module, on board computer Raspberry Pi and telemetry radio signal etc.

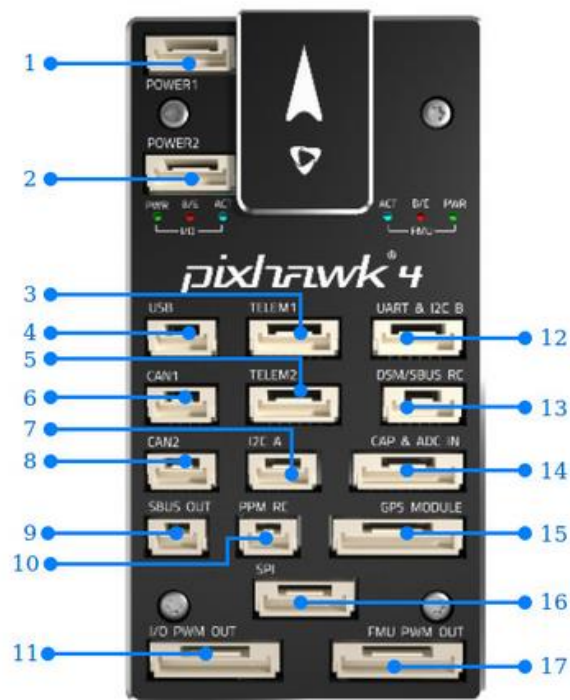


Figure 4.2 (A)

Fig has following ports:

1. Power module 1
2. Power module 2
3. Telemetry 1 (radio telemetry)
4. USB
5. Telemetry 2 (companion computer)
6. CAN1 bus
7. I2C
8. CAN2
9. S.BUS out for S.BUS servos
10. Radio Control Receiver Input (PPM)
11. Main outputs
12. UART and I2C
13. Radio Control Receiver Input (DSM/SBUS)
14. Input Capture and ADC IN
15. GPS module
16. SPI bus
17. AUX outputs (FMU PMU out)

4.2.3 4500 mAh Li-PO Power Battery



Figure 4.3

The system uses a 4500 45C mAh Li-Po battery which has a capacity of 4500 mAh and 45C discharging rate. The battery allows for 30-35 minutes flight time without any payload which is sufficient to complete the test.

4.2.4 Radio Control Transmitter and receiver



Figure 4.4

4.2.5 Transceiver Telemetry Radio



Figure 4.5

4.2.6 Microprocessor Raspberry Pi 4



Figure 4.6

The system uses the Raspberry Pi 4 computer which has 64-bit quad-core processor, up to 4 GB of RAM, dual-band 2.4/5.0 GHz wireless LAN, Bluetooth 5.0, USB 3.0, 2 x micro-HDMI ports, 40 Pin GPIO header, Power over Ethernet (PoE) enabled and micro SD card slot for loading operating system and data storage. The micro computer can be operated using different software such as Raspbian, Linux and Windows IoT. The Raspberry Pi 4 is using Ubuntu 18.04 version. The weighs of Raspberry Pi 4 is 45 grams which is suitable to use for an UAV.

4.2.7 Raspberry Pi Camera

The Raspberry Pi camera module version 2 is 8 megapixel focus camera which supports 1080p, 720p and 360p. The camera dimensions are 25 x 24 x 9 mm and it attached via ribbon cable to the CSI port of Raspberry Pi 4. To capture frames for image processing, the camera can be accessed using the Pi camera Python module or the ROS environment.

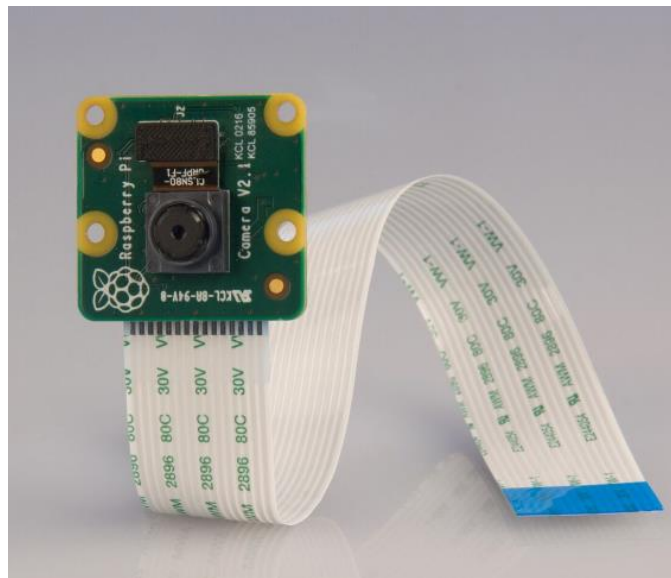


Figure 4.7

4.2.8 StromPi Power and Battery Module

The StromPi enables the Raspberry Pi 4 with an optional plug-in battery unit, which adds a rechargeable emergency power source to Raspberry pi 4 with a LiFePo4 battery. StromPi 3 allows the voltage between 6-61V and a current with up to 3A.

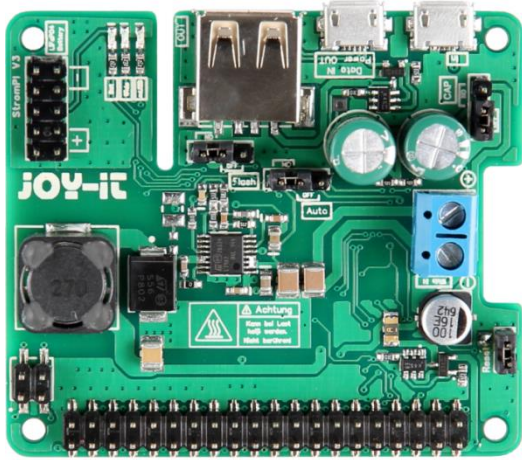


Figure 4.8 (A)



Figure 4.1 (B)

4.2.9 Intel Real Sense Depth Camera

Intel RealSense Depth Camera has used to navigate the UAV according to AprilTag fiducial marker. The working of this camera has explained in next chapter. This camera combines the robust depth sensing capabilities of the D435 with the addition of an inertial measurement unit (IMU).



Figure 4.9

4.2.10 WiFi Connection

A wifi network is used to connect the on board microcomputer i.e. Raspberry Pi 4 with Mission Planner software in order to give commands via SSH connection.

4.2.11 Mission Planner Software

The Mission Planner software is used on the ground station for this system to control the UAV. For autonomous vehicle, Mission Planner can be used as a configuration tool or as a dynamic control supplement. It's a open source application which is developed by Michael Osborne for APM autopilot project. It can be used to load the firmware into Pixhawk FCU and setup, configure the UAV. Mission Planner can allow monitoring the UAV status while in working mode with telemetry radio signal that is connected to Mission Planner. Fig shows the UAV status on map (green color), alarm status and some small windows with having values of Attitude, Yaw etc.



Figure 4.10

4.2.12 3D Printing for Raspberry Pi 4 microcomputer