1.INTRODUCTION

The Internet of Things (IoT) is becoming a major paradigm, enabling applications in industries from smart cities to health care and dietary assessment. IoT devices embed sensors and low-power processors into the physical world, allowing a rich set of information to be gathered by a diverse set of devices. IoT devices are becoming so common that they are expected to make up 18 billion of the 29 billion connected devices worldwide by 2021. The widespread deployment of IoT devices necessitates that The system understands the challenges of designing solutions for these next-generation platforms. IoT devices are on the cusp of a revolution as they start to embrace some form of intelligence via cognitive or deep learning computing capability to provide intelligent end user services. For instance, smart swim watches seek to distinguish between swim strokes, and digital home assistants seek to understand human voices and language. Drones are an emerging form of new IoT devices, flying in the sky with full network connectivity capabilities. Intelligent drones with cognitive computing skills need the capability to automatically recognize and track objects to free users from the tedious task of controlling them, all of which must be performed within the power constrained environment of a Li-Po battery.

In this system, there is a cognitive drone platform, a domain refers to as the Flying IoT, to quantify the power and performance characteristics of cognitive applications on these emerging mobile devices. Specifically, the system has an application called Follow the Leader, which automatically detects, tracks, and follows a moving human target. The application is centered on a machine learning task called object detection, which is its most computationally intensive kernel. The ability to perform basic computer vision tasks like object detection is a necessary step toward new and intelligent applications such as sports photography and package delivery. Cognitive applications are often very computationally intensive, which makes them difficult to run on embedded computers that have low-power, lightweight, small-size design requirements. The challenge of performing object detection on drones is to balance performance and power efficiency. To operate successfully, Follow the Leader must detect a person multiple times per second, or it could lose the person it is attempting to track. This real-time performance requirement is difficult to satisfy on extremely low-power CPUs, or even GPUs, even for simple shallow machine learning models.

For complex multiclass deep models like convolutional neural networks (CNNs), desktop or server-level processors are required. So, extremely low-power, low-performance processors alone are not sufficient without hardware specializations, but hardware specialization introduces design complexity and nonrecurring engineering costs. As an alternative to high-performance, low-power hardware specialization approaches, the system investigate a general-purpose software paradigm to sustain low power consumption and small processor form factor.

Using a range of off-the-shelf low-power to high-performance processors, the system show that it is impractical tom simultaneously achieve real-time performance and low power when executing a cognitive drone application. Therefore, the system proposes a sensor-cloud architecture to partition data collection and processing between the edge and the cloud. The system characterizes a drone application that runs on the i.MX6, a low-power, low performance system, and on the TX1, a high-power, high-performance system. Both of these computational systems are typically found on existing drone platforms. Neither can provide the necessary performance and energy efficiency to make our application viable. Subsequently, the system characterizes the effects that incorporating the cloud has on the performance, power, and energy consumption of the drone application on these platforms. The system shows that by offloading complex object-detection models to the cloud, the system can improve performance while minimizing edge power to negligible levels. The sensor-cloud architectures sustained low power on an energy constrained drone platform, while delivering the real-time performance needed by continuous object detection and decision making. Finally, the system demonstrates how various common software-level optimizations, such as image down sampling and lossy compression, can trade small accuracy loss for significant performance and energy efficiency improvements.

2.LITERATURE SURVEY

2.1 A Survey on Sensor-Cloud: Architecture, Applications and Approaches

Many organizations desired to operate their businesses, works and services in a mobile, dynamic, and knowledge-oriented fashion. Activities like e-learning, environmental learning, remote inspection, health-care, home security and safety mechanisms etc. requires a special infrastructure that might provide continuous, secured, reliable and mobile data with proper information knowledge management system in context to their confined environment and its users. An indefinite number of sensor networks for numerous healthcare applications has been designed and implemented but they all lacking extensibility, fault-tolerance, mobility, reliability and openness. Thus, an open, flexible and rearrange able infrastructure is proposed for healthcare monitoring applications. Where physical sensors are virtualized as virtual sensors on cloud computing by this infrastructure and virtual sensors are provisioned automatically to end users whenever they required. In this paper The system reviewed some approaches to hasten the service creations in field of healthcare and other applications with Cloud-Sensor architecture. This architecture provides services to end users without being worried about its implementation details. The architecture allows the service requesters to use the virtual sensors by themselves or they may create other new services by extending virtual sensors.

Sensors are capable of sensing the several appearances and can be utilized in several areas like healthcare, defense, government services, environmental services etc. These sensors may provide various useful data but are closely attached to each of their relevant applications and services directly, causing several other services to be unused. Hence large number of our meaningful, expensive resources may become waste. But if anyhow The system can integrate these sensors by sharing each other’s valuable data through the number of unlimited services, it would accelerate the service creation. To realize this sensor cloud infrastructure has been proposed which is the extended form of cloud computing to manage our valuable sensors scattered around the network (sensor). This infrastructure would provide the service instances (virtual sensors) automatically to users as and when requested in same way as these virtual sensors are part of IT resources (like disk storage, CPU, memory etc.) to the end users. Before generating the service instances; the IT resources (like CPU, Storage devices etc.), sensor capable devices, service templates (that has to be used to create virtual sensors) should be prepared first. Users request for service instances according to their needs by selecting an appropriate service template which will then provide the service instances freely and automatically because of cloud computing services integration. Once service instances became useless, it can be deleted quickly by users to avoid the utilization charges for these resources. Sensor service provider will manage the service templates and can add or delete the new service template as and when the requirement for template is needed by applications and services. Automation of services played a vital role in provisioning of cloud computing services and automation can cause the delivery time of services to be better. Before the emergence of cloud computing, services were provided by human influence and the performance metrics like efficiency, flexibility, delivery time etc. would have experienced an adverse effect on the system. But the cloud computing service model has reduced the cost expenses, delivery time and has also improved the efficiency and flexibility because the service providers need not to worry about preparing the IT resources and its infrastructure.

Cloud-Sensors can be used for deploying the health related applications like chronic patients with heart problems, blood sugar, sleep activity pattern monitoring, respiratory conditions, diabetics, cardio-vascular disease etc. Earlier the trials of individual's data like level of blood sugar, weight, heart rate, pulse rate etc. are reported everyday through some telemedicine interface. The patient's trial information sends to a dedicated server and is stored there for doctors or helpers to analyze it sometime later. This system has some level of adversity when the patient moves on from its current location i.e., when patient is "on the go". Thus there needed a more progressive, rapid, and mobile approach where our data can be processed in pipelined and parallel fashion, thereby making the system easier to scale and cost effective in terms of the resources available. The pipeline processing of data sets or instructions enables the overlapped operations into a conceptual pipe with all the stages of pipes processing simultaneously but handling of sensor data stream is not that straight forward and will be dependent on the nature of algorithm.

2.2 Internet of Things (IoT) based Sensors to Cloud system using Arduino Due

The system proposed in this paper is an advanced solution for monitoring the weather conditions at a particular place and make the information visible anywhere in the world. The technology behind this is Internet of Things (IoT), which is an advanced and efficient solution for connecting the things to the internet and to connect the entire world of things in a network. Here things might be whatever like electronic gadgets, sensors and automotive electronic equipment. The system deals with monitoring and controlling the environmental conditions like temperature, relative humidity, light intensity and sound level with sensors and send this information to the cloud and then plot the sensor data as graphical statistics. The data updated from the implemented system can be accessible in the internet from anywhere in the world

Internet of Things where 'things'- sensors and devices transmit data directly to the Internet has become an enabling technology eco-system with several application areas are Smart Home, Smart Farming, Smart Grid, Industrial Internet, Connected Health, Smart Supply Chain etc. The application list is impressive, however, since the technologies involved are many- sensors, microcontrollers, wireless networking, cloud based services, mobile apps, web pages -practical implementation of an IoT application is complex. Present innovations in technology mainly focus on controlling and monitoring of different activities. These are increasingly emerging to reach the human needs. Most of this technology is focused on efficient monitoring and controlling different activities. The application list is impressive, however, since the technologies involved are many-sensors, microcontrollers, wireless networking, cloud based services, mobile apps, web pages -practical implementation of an IoT application is complex.

Present innovations in technology mainly focus on controlling and monitoring of different activities. These are increasingly emerging to reach the human needs. Most of this technology is focused on efficient monitoring and controlling different activities. An efficient environmental monitoring system is required to monitor and assess the conditions in case of exceeding the prescribed level of parameters (e.g., noise, CO and radiation levels).

When the objects like environment equipped with sensor devices, microcontroller and various software applications becomes a self-protecting and self-monitoring environment and it is also called as smart environment. In such environment when some event occurs the alarm or LED alerts automatically. The effects due to the environmental changes on animals, plants and human beings can be monitored and controlled by smart environmental monitoring system. By using embedded intelligence into the environment makes the environment interactive with other objectives, this is one of the application that smart environment targets.

Thing Speak is a IoT cloud service provider. Embedded IoT devices like Arduino, Raspberry Pi can be connected to internet. These boards then can fetch data or upload data to Thing Speak storage using APIs. The data stored by a device can be accessed by other client entities like Mobile, Tablet, laptop connected to internet using Thing Speak APIs. So in short Thing Speak is an IoT service provider that provides APIS to upload, retrieve and visualize data from IoT devices over cloud.

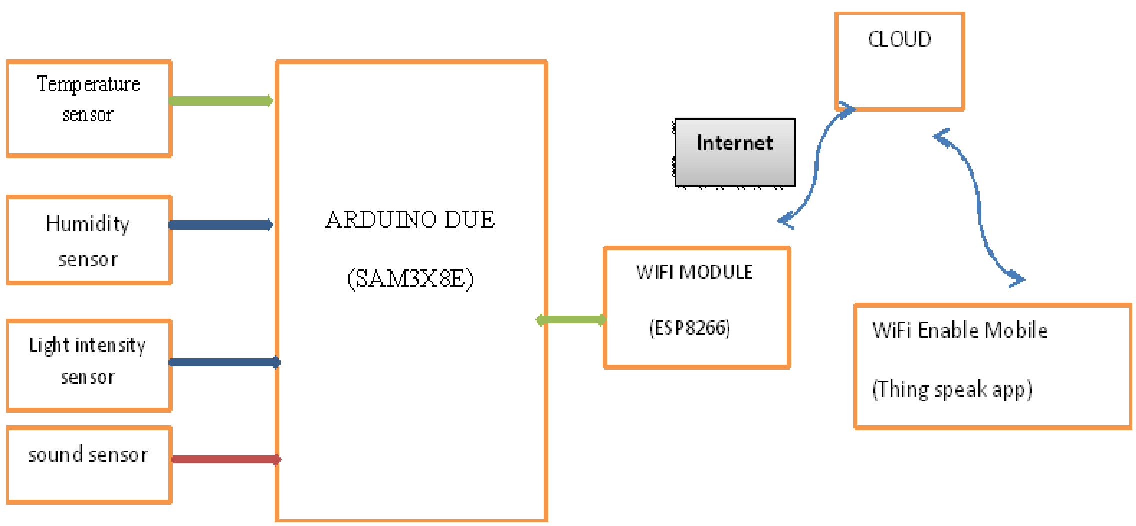

Fig 2.1 IoT based Sensors to Cloud system using Arduino Due

2.3 Flying Objects Detection from a Single Moving Camera

This system proposes an approach to detect flying objects such as UAVs and aircrafts when they occupy a small portion of the field of view, possibly moving against complex backgrounds, and are filmed by a camera that itself moves. Solving such a difficult problem requires combining both appearance and motion cues. To this end The system proposes a regression-based approach to motion stabilization of local image patches that allows us to achieve effective classification on patio-temporal image cubes and outperform state of- the-art techniques. As the problem is relatively new, the system collected two challenging datasets for UAVs and Aircrafts, which can be used as benchmarks for flying objects detection and vision guided collision avoidance. The system is headed for a world in which the skies are occupied not only by birds and planes but also by unmanned drones ranging from relatively large Unmanned Aerial Vehicles (UAVs) to much smaller consumer ones. Some of these will be instrumented and able to communicate with each other to avoid collisions but not all. Therefore, the ability to use inexpensive and light sensors such as cameras for collision-avoidance purposes will become increasingly important.

This problem has been tackled successfully in the automotive world and there are now commercial products designed to sense and avoid both pedestrians and other cars. In the world of flying machines most of the progress is achieved in the accurate position estimation and navigation from single or multiple cameras while not so much is done in the field of visual-guided collision avoidance On the other hand, it is not possible to simply extend the algorithms used for pedestrian and automobile detection to the world of aircrafts and drones, as flying object detection poses some unique challenges. As a result, motion cues become crucial for detection. However, they are difficult to exploit when the images are acquired by a moving camera and feature backgrounds that are difficult to stabilize because they are non-planar and fast changing. Furthermore, since there can be other moving objects in the scene, in these situations, state-of-the-art techniques that rely on either image flow or background stabilization lose much of their effectiveness. In this paper, the system detects whether an object of interest is present and constitutes a danger by classifying 3D descriptors computed from spatio-temporal image cubes. The system will refer to them as st-cubes.

These st-cubes are formed by stacking motion-stabilized image windows over several consecutive frames, which gives more information than using a single image? What makes this approach both practical and effective is a regression-based motion-stabilization algorithm. Unlike those that rely on optical flow, it remains effective even when the shape of the object to be detected is blurry or barely visible. St-cubes of image intensities have been routinely used, for action recognition purposes using a single fixed camera. In contrast, most current detection algorithms work on a single frame, or integrate the information from two of them, which might not be consecutive, by taking into account optical flow from one frame to another. Our approach can therefore be seen as a way to combine both the appearance and motion information to achieve effective detection in a very challenging context.

2.4 Optimization of Energy Consumption of Drones with Software-Based Flight Analysis

The capabilities and applications of new drones require a high demand of power that traditional batteries cannot sustain, triggering the need of developing strategies to promote efficient energy usage on drone platforms. However, to reduce the energy consumption it is required to have reliable means to measure the device's behavior and its relationship to the battery discharge. This paper introduces Green Flight, a software system that acquires data during a drone mission, featuring an online battery discharge analyzer and a real-time suggestions mode to help to optimize the overall power consumption of the drone, allowing for longer, more autonomous missions. The fast development and diffusion of the new generation of unmanned aerial vehicles, more popularly known as drones, has brought a number of needs and opportunities for different research paths. The drone industry has been a hot topic for the past several years, and the focus of many discussions in the technology sector in different aspects: applications, handling, management, airworthiness, autonomy, and energy efficiency. In this regard, the capabilities of new generation drones boost a high demand of power that traditional batteries cannot sustain.

This situation triggers the need of strategies to promote efficient energy usage to reduce the power demand on drone platforms, with the goal of guaranteeing the capacity of the drone to complete a mission safely. In the past, several research and practitioner work has focused towards designing autonomous, energy efficient targets (for instance, smartphones, tablets, or wearables) however, there is little research and experience applying these approaches in drone platforms. Moreover, an important need that arises when designing energy aware systems is the ability to understand the way in which the energy is used and spent. This opens the doors to design, implement and validate reliable means to measure the battery discharge for drone targets. In this paper, the system outlines a strategy to measure the energy consumption of commercial quad drones. Our strategy includes a systematic data collection of drone missions, and online data analysis to characterize and report out the actions that relate the behavior of the drone to the overall consumption of energy.

In this way, users may identify what are the maneuvers, movements and general aspects of the flight profile (load, altitude, speed) that can involve more energy investment, and as a consequence faster battery discharge that leads to shorter drone autonomy. This knowledge can be of utmost importance for strategic actions, such as: identifying and monitoring power-relevant characteristics of drones; proposing metrics to evaluate the energy performance of the drone; and establishing baselines to manage energy consumption and identify energy relevant faults.

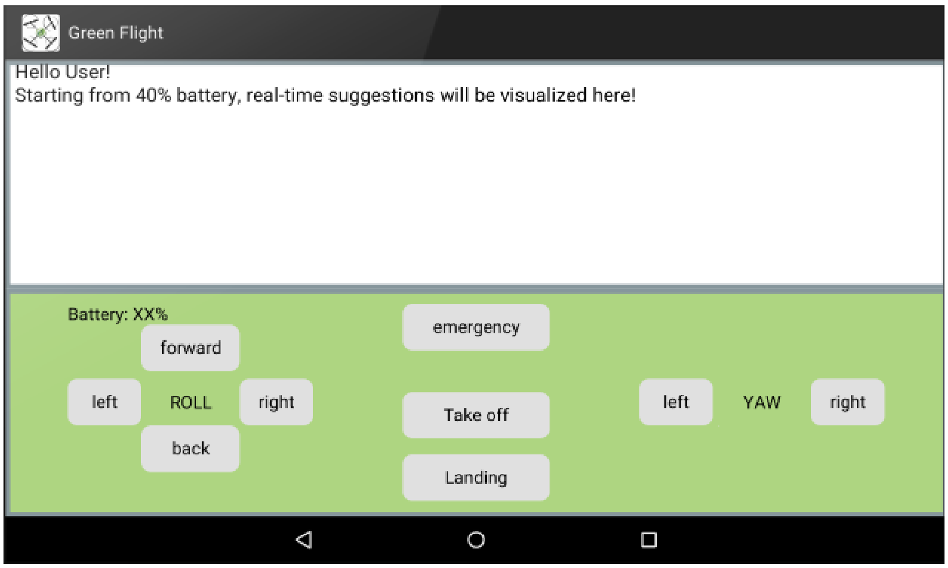

Fig 2.2 Green flight application

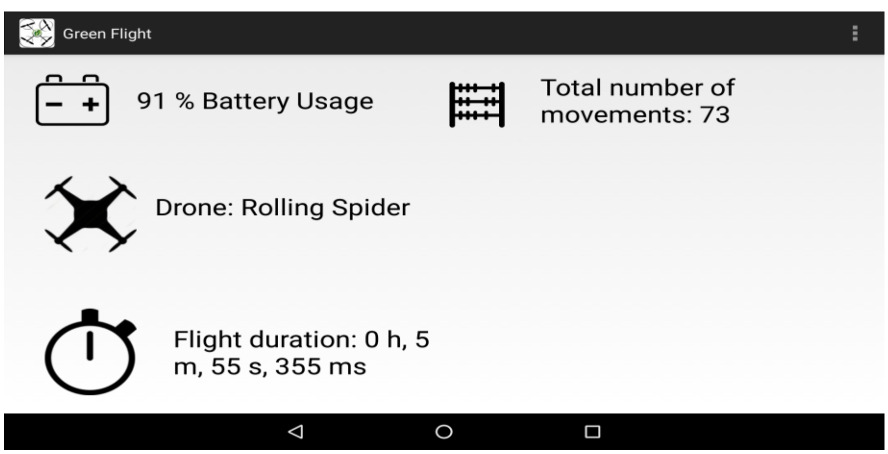

Fig 2.3 Information from Green flight

2.5 Understanding Autonomous Drone Maneuverability for Internet of Things Applications

Increasing sensing and communication capabilities combined with falling prices have made drones very attractive for Internet of Things applications. A key requirement of these applications is that the drones should be autonomously maneuvered by computer programs. It is therefore important to understand the practical limitations of autonomous drone maneuverability to ensure that target application performance is met. In this paper, the system first analyzes drone maneuverability using theory to shed light on the tradeoff between the flying speed and the turning agility of the drone. To investigate the practical maneuverability performance, the system then emulates as well as fly a commercial drone under the control of an Android program. The system reveals some practical maneuverability factors that must be considered for the applications that require frequent changes of direction for the drone.

According to Federal Aviation Administration, drone sales are expected to grow from 2.5 million in 2016 to 7 million in 2020. This massive popularity of drones is driven by the increasing sensing and communication capabilities combined with their falling prices. Modern drones are now equipped with a large number of sensors, such as temperature, humidity, air pollution, GPS, 4K camera, etc., as well as an expanding range of wireless communication capabilities including WiFi, Bluetooth, 4G, RFID, and so on. Finally, drones have also improved in mechanical performance, such as flying speed, degree of autonomy, agility and maneuverability, and are available in a variety of form factors. These developments have made drones a very attractive platform to launch many Internet of Things (IoT) applications. For example, a drone can be instrumented with the latest IoT gateway technologies, such as LoRa to read gas, water, and power meters from the sky, giving an unprecedented advantage to IoT operators. A drone equipped with the right sensors can be used to collect specific air quality data from hard to reach places in a smart city and transmit that data to an IoT server in real-time using LoRa or even 4G depending on the availability. Indeed, researchers have already started visionary projects that use drones as the main IoT sensing and data delivery platform.

In most of the envisaged IoT applications, the drone has to be flown autonomously using computer algorithms that can quickly and efficiently maneuver it to the right location at the right time. Any small maneuverability error at any stage of a complicated flight can propagate quickly through the rest of the flight causing significant performance issues for the target application. It is therefore important to understand the real effect of a particular command specified in the application program interface (API) of the drone. The goal of this study is to experiment with a real commercial drone to understand these API commands for drone maneuverability and identify any factors that must be considered when designing autonomous drone maneuvering. In particular, the system wants to answer questions like: what is the tradeoff between flying speed and turning agility, what is the impact of flying speed on the battery life, how frequently The system can make a drone change its direction, and so on. The system has attempted to answer these questions using both theory as well as experiments with a DJI Phantom drone. Our experiments reveal some key practical maneuverability factors that have not been captured in the basic theoretical formula available to study drone trajectories.

3.TECHNOLOGIES USED

3.1 Object Detection

The system characterizes the performance and power of different processors running Follow the Leader on the edge. The system shows that edge processors cannot achieve real-time performance for this application without consuming excessive amounts of power. Drones typically have quite constrained battery capacities, which severely limits their flight time. Excessive power consumption by processors can reduce the amount of time they are in the air even further. Software optimizations such as down sampling can improve performance and power but still fall far short of satisfying the power and performance goal. When running on the drone, Follow the Leader's workflow can be broken into several pipeline stages: fetching images, preprocessing images, detecting objects in those images, and taking action to move the drone toward detected targets. These stages, taken together, constitute a single frame of the application, and The system quantify the performance of our application by the number of frames that are executed every second. The i.MX6 on the drone is a low-power, low-performance chip typical of battery constrained Flying IoT devices. The system characterizes its power and performance and show that although the i.MX6 satisfies a low-power device's power budget, its performance is far too low to provide real-time cognition.

Ultra-low-power CPU architectures like the i.MX6 do not satisfy the performance needs of real-time object detection. Therefore, The system explore a higher performance embedded system on chip (SoC) as an alternative and analyze the power and performance tradeoffs. The system run the object-detection algorithms on the Nvidia Jetson TX1 because its GPU provides higher performance for these applications and is more power efficient than a CPU. At the time the research was conducted, the Nvidia Jetson TX2 platform was yet to be released.

3.2 Jetson TX1

NVIDIA Jetson with GPU-accelerated parallel processing is the world's leading embedded AI computing platform. The Jetson portfolio of devices, featuring the new NVIDIA Jetson TX2, delivers more performance and features for Artificial Intelligence at the edge. Devices at the edge, from drones to intelligent cameras, need on-board AI to process complex data without relying on network connectivity. AI at the Edge is the future of industry, transforming processes in manufacturing, industrial inspection, agriculture, general robotics, security, and AI cities. The world's first supercomputer on a module, Jetson TX1 is capable of delivering the performance and power efficiency needed for the latest visual computing applications. It's built around the revolutionary NVIDIA Maxwell™ architecture with 256 CUDA cores delivering over 1 TeraFLOPs of performance. 64-bit CPUs, 4K video encode and decode capabilities, and a camera interface capable of 1400 MPix/s make this the best system for embedded deep learning, computer vision, graphics, and GPU computing.

3.3 IMX6

The i.MX range is a family of Freescale Semiconductor (now part of NXP) proprietary microcontrollers for multimedia applications based on the ARM architecture and focused on low-power consumption. The i.MX application processors are SoCs (System-on-Chip), that integrate many processing units into one die, like the main CPU, a video processing unit and a graphics processing unit for instance. The i.MX products are qualified for automotive, industrial and consumer markets. Most of them are guaranteed for a production lifetime of 10 to 15years Many devices use i.MX processors, such as Ford Sync, Kobo eReader, Amazon Kindle, Sony Reader, Onyx Boox readers/tablets, Solid RunSOM's (including CuBox), some Logitech Harmony remote controls and Squeezebox radio, some Toshiba Gigabeat mp4 players. The i.MX range was previously known as the "Dragon Ball MX" family, the fifth generation of Dragon Ball microcontrollers. i.MX originally stood for "innovative Multimedia eXtension".

3. 4 Cloud Server

The system envisions future cloud computing systems to support drone services. The cloud server The system use is equipped with a 3.6 GHz Intel i7 processor and a GeForce GTX 1080 Ti graphics card and is connected to the drone through Wi-Fi. In the future, emerging 5G networks are expected to guarantee Wi-Fi speeds6 while becoming more energy efficient than 4G mobile data networks.7 To run our cognitive workloads, The system installed our server with CUDA 8.0, cudNN v5.1, OpenCV 3.1, and the Caffe deep learning framework.

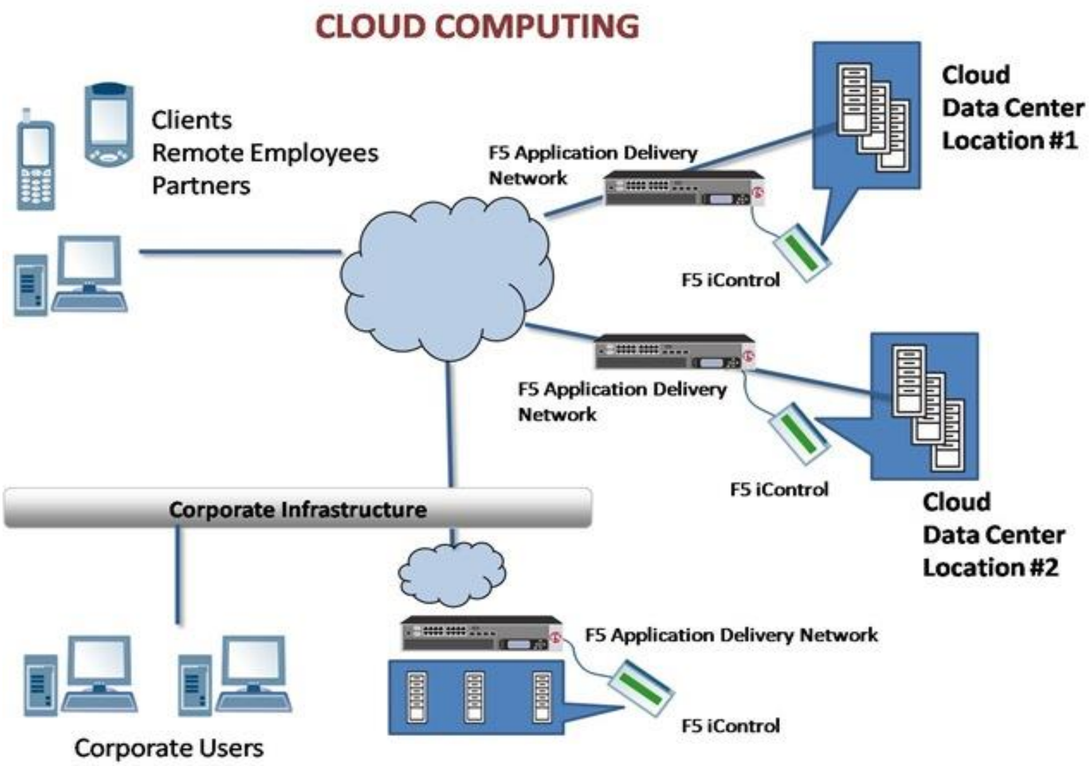

Fig.3.1 Cloud computing

3. 5 Sensor-Cloud Architecture

The system proposes a sensor-cloud system to bring server-level computational capability to low-power IoT devices such as drones. In a sensor-cloud system, computationally intensive tasks are offloaded to the cloud while data collection tasks are done at the edge. The system modifies our Follow the Leader application so that it offloads object detection, our bottleneck stage, to the cloud, enhancing our application's performance while maintaining a low power consumption. In fact, with software optimizations such as compression, even low-performance CPUs like the i.MX6 can achieve the performance of the TX1 on the edge. The sensor-cloud application's workflow is represented by the following pipeline stages: the drone fetches images, preprocesses them, compresses them, and transmits them to the cloud. The cloud then decompresses the images, analyzes them using object-detection algorithms, and sends the results back to the drone.

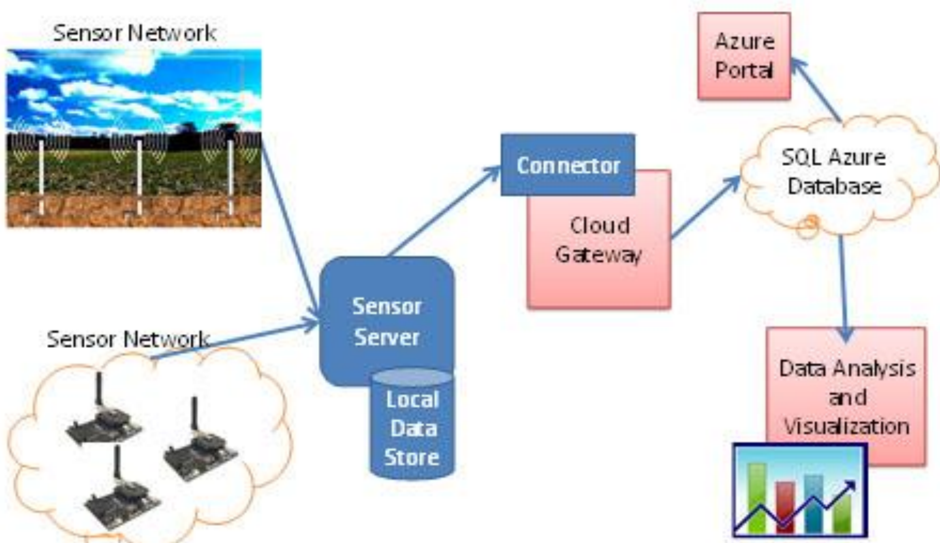

Fig 3.2 Sensor Cloud Architecture

A wireless sensor network (WSN) is a wireless network consisting of spatially distributed autonomous devices using sensors to monitor physical or environmental conditions. A WSN system incorporates a gateway that provides wireless connectivity back to the wired world and distributed nodes. The wireless protocol you select depends on your application requirements. Some of the available standards include 2.4 GHz radios based on either IEEE 802.15.4 or IEEE 802.11 (Wi-Fi) standards or proprietary radios, which are usually 900 MHz Finally, the drone takes action based on the response from the cloud. Real-time applications such as Follow the Leader cannot tolerate long communication delays between drones and the cloud. Thus, our performance characterization takes into account the time delays associated with network communication in the application. One assumption The system makes is that there is no failure in network connectivity.

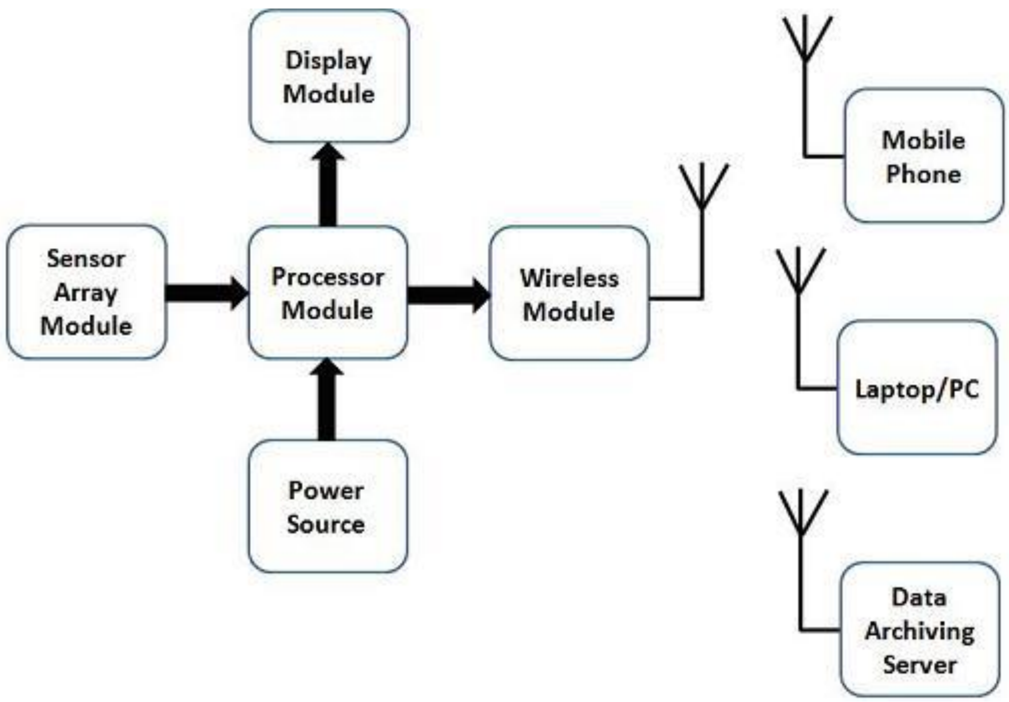

Fig 3.3 Wireless Sensor Network

3.6 Drone Computing Platform

For a typical Flying IoT computing platform, the system uses the i.MX6, a low-power, low-performance ARM Cortex A-9 processor that comes with the 3DR Solo, a commercially available state-of-the-art hobbyist drone. The 3DR Solo is designed in a fashion typical to many autonomous drones, and its processor is indicative of the computational performance one might expect to find on a typical Flying IoT platform. The i.MX6 is powerful enough to enable only simple flight operations such as instructing a drone to start taking pictures once it has reached a certain height. To evaluate higher-performance edge processors, the system also attaches a state-of-the-art embedded.

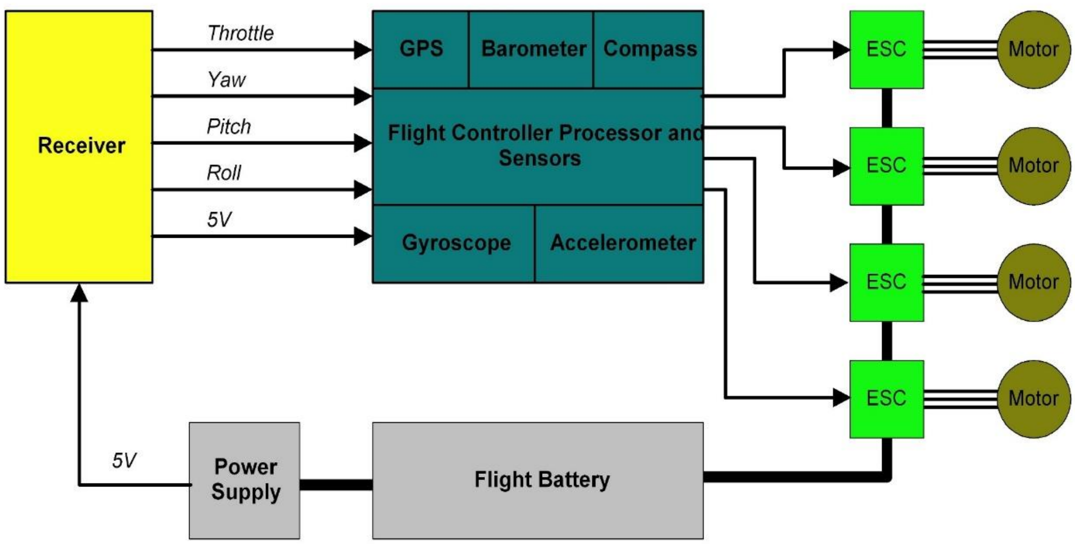

Fig3.4 Drone Architecture

4.WORKING

4.1 Follow The Leader

The system creates a *Follow the Leader* application where our drone detects and follows a moving human target in real time. Many smart drones, in domains from security to sports photography, must be capable of following human targets, whether to record video of a quickly moving athlete or to monitor a suspicious individual in a crowd. To detect targets, our application runs object-detection algorithms on images taken from the drone's cameras. The drone then flies along the horizontal plane, centering its target in the middle of the drone's field of view. An autonomous drone application has strict real-time requirements, because it must fetch, analyze, and react to sensory data quickly enough to avoid its moving targets from exiting its field of view. In this work, the system set a real-time performance goal of 10 frames per second (fps).

Fig.4.1 Follow the Leader

4. 2 Dataset

To ensure a controlled test environment, the system evaluate the drone application indoors, with the drone stationary, replacing its camera input with “positive test” images from the INRIA Person 10 The drone reads images from the dataset and treats them as inputs from its own camera, attempting to fly toward the people in those images. In practice, the application’s image inputs would be dynamically changing, whereas the images from the INRIA dataset are static. Regardless, the proposed evaluation model is sufficient, because the application’s performance does not change based on whether or not the images it is operating on are related to the images that came before. The object-detection algorithms, for example, run at the same speed regardless. One limitation of using a static image dataset is that it does not precisely represent the performance of a physical camera. To overcome this issue, the system converts the images in the dataset to the BMP format for experiments involving the i.MX6, and The system keep the original PNG format for experiments involving the TX1. This minimizes the difference between the time spent decoding images from the dataset and the time that would have been spent fetching images from physical cameras to within about 0.04 seconds.

4.3 Cognitive algorithms.

To allow our Follow the Leader application to detect targets, the system evaluate both multiclass and single-class object-detection algorithms. Multiclass object detectors can detect multiple different types of objects in a single image, whereas single-class object detectors are designed to detect only a single type of object at a time. The system picks two state-of-the-art CNN multiclass object detectors that are trained end-to-end from raw image pixels—Faster R-CNN and YOLO—and two single-class object detectors that are trained using hand-crafted features— Haar cascade classifiers and histogram of oriented gradients (HOG) detectors.

For Faster R-CNN, the system use the official Python implementation along with the pre trained model, ZF. For YOLO, The system use the official implementation and the pre trained model trained with the COCO dataset. For the Haar and HOG single-class object detectors, The system used the CUDA implementation in OpenCV 2.4.13 when GPUs were available, and the CPU implementation when they were not. Methods to improve CNN efficiency, such as pruning, are outside the scope of this article; our focus, instead, is on evaluating the performance and power of preexisting models and algorithms.

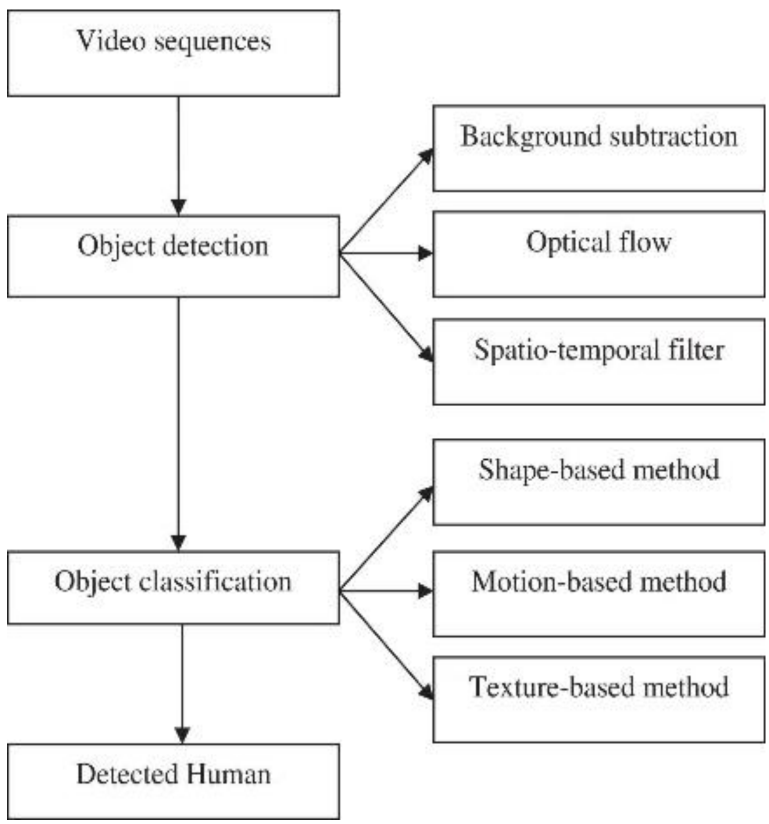

Fig 4.2 Single-class object-detection

4.4 YOLO Algorithm

YOLO is a new approach to object detection. Prior work on object detection repurposes classifiers to perform detection. Instead, the system frame object detection as a regression problem to spatially separated bounding boxes and associated class probabilities. A single neural network predicts bounding boxes and class probabilities directly from full images in one evaluation. Since the whole detection pipeline is a single network, it can be optimized end-to-end directly on detection performance. In this unified architecture is extremely fast. The base YOLO model processes images in real-time at 45 frames per second. A smaller version of the network, Fast YOLO, processes an astounding 155 frames per second while still achieving double the mAP of other real-time detectors. Compared to state-of-the-art detection systems, YOLO makes more localization errors but is far less likely to predict false detections where nothing exists. Finally, YOLO learns very general representations of objects. It outperforms all other detection methods, including DPM and RCNN, by a wide margin when generalizing from natural images to artwork on both the Picasso Dataset and the PeopleArt Dataset

4.5 HAAR Cascade Classifiers

The core basis for Haar classifier object detection is the Haar-like features. These features, rather than using the intensity values of a pixel, use the change in contrast values between adjacent rectangular groups of pixels. The contrast variances between the pixel groups are used to determine relative light and dark areas. Two or three adjacent groups with a relative contrast variance form a Haar-like feature. Haar-like features, Haar features can easily be scaled by increasing or decreasing the size of the pixel group being examined. This allows features to be used to detect objects of various sizes.

4.6 HOG Algorithm

The histogram of oriented gradients (HOG)is a feature descriptor used in computer vision and image processing for the purpose of object detection. The technique counts occurrences of gradient orientation in localized portions of an image. This method is similar to that of edge orientation histograms, scale-invariant feature transform descriptors, and shape contexts, but differs in that it is computed on a dense grid of uniformly spaced cells and uses overlapping local contrast normalization for improved accuracy

4.7 R CNN Algorithm

R-CNN is a state-of-the-art visual object detection system that combines bottom-up region proposals with rich features computed by a convolutional neural network. At the time of its release, R-CNN improved the previous best detection performance on PASCAL VOC 2012 by 30% relative, going from 40.9% to 53.3% mean average precision. Unlike the previous best results, R-CNN achieves this performance without using contextual rescoring or an ensemble of feature types.

4.8 Power and Energy

In addition to improving performance, switching to a sensor-cloud system can potentially reduce power and energy consumption at the edge. On the i.MX6, the instantaneous power consumption falls only slightly, but the performance improvement brought by the cloud causes our energy consumption per frame to plummet to below 1 J per frame. The TX1's instantaneous power consumption falls by 0.5 to 2 W for the Haar, HOG, and YOLO detectors, and by a full 6.4 W for Faster R-CNN, because the application no longer uses the GPU. These instantaneous power savings help reduce the energy consumed per frame for all detectors, but especially for YOLO and Faster R-CNN, which consume 6 to 11 times less energy to process frames. Typically, in a sensor-cloud system, GPUs would not be installed on the edge. The system can see from these results that utilizing GPUs in the cloud instead of on the edge can yield significant power and energy savings.

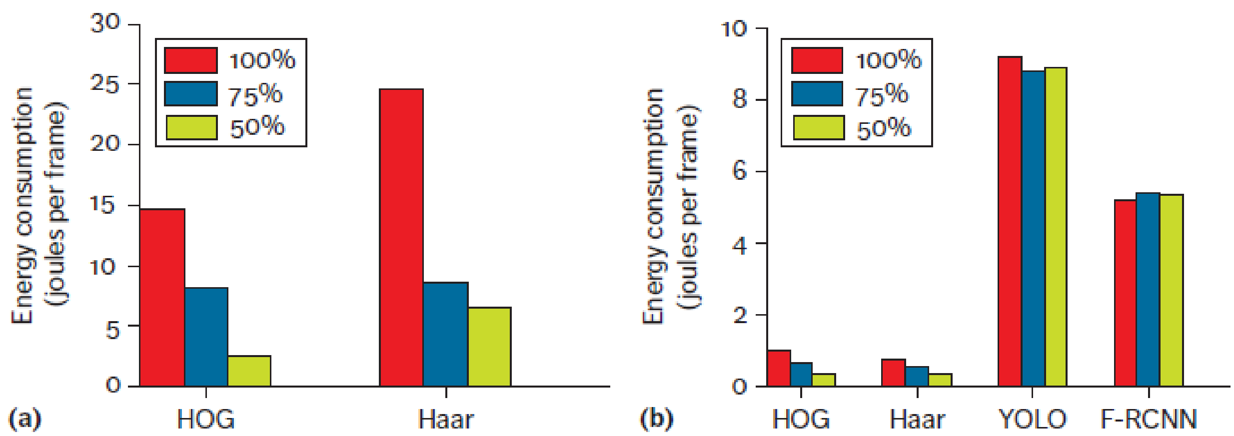

Fig 4.3 Energy consumption per frame: (a) i.MX6 and (b) TX1. The scaling factors are shown in the figure, where 100 percent means the original figure size is not scaled down. Note that the *y*-axis of (a) is scaled to three times the *y*-axis of (b).

5.ADVANTAGES AND DISADVANTAGES

The motor, sensors, controls used in the drones are highly efficient. Here the datas are collected by the sensors are stored in the cloud server and from there it will decide how it should be worked. Since the system uses the comparison of two processors such as tx1 and imx6, the power consumed is low. Using of the cognitive algorithm the system gives high quality data processing. The time taken for working of the system based on the sensing is very quick. Here the drones also detect and follow moving human target in real time.

Here the multiple human detection is less efficient. The speed and performance of the system completely based on the network system there for if the network is slow then the whole system performance will become slow. The cost as well as the implementation of the system is very difficult.

6.FUTURE ENHANCEMENTS

The performance of sensor-cloud applications is significantly limited by the speed at which they can compress and transmit data. Currently, compression on drone processors is typically done by CPUs. However, by developing specialized compression accelerators, researchers can alleviate the pressure put on CPUs, dramatically improving performance. Furthermore, it will be important to investigate the implications of network stability on drone applications that utilize the cloud. Our study uses a Wi-Fi network to connect the drone to the cloud, but it will be worthwhile to look into the impact of 4G and future 5G networks on the speed and energy efficiency of drone applications.

7.CONCLUSION

The Internet of Things is entering a new paradigm where devices on the edge need both cognitive capability and the ability to interact directly with their environments in real time. Although our work demonstrates that sensor-cloud architectures can accelerate Flying IoT applications to near-real-time performance, it also exposes some of the challenges associated with them. The performance of sensor-cloud applications is significantly limited by the speed at which they can compress and transmit data. Currently, compression on drone processors is typically done by CPUs. However, by developing specialized compression accelerators, researchers can alleviate the pressure put on CPUs, dramatically improving performance.

With appropriate compression and down sampling optimizations, sensor-cloud architectures can reduce edge power significantly compared to placing all the computation on the edge. Meanwhile, sensor-cloud architectures enhance application performance to almost the level of the real-time target by offloading computationally intensive software kernels to the cloud.

REFERENCES

[1.] W.F. Domoney et al., "Smart City Solutions to Water Management using Self-Powered, Low-Cost, Water Sensors and Apache Spark Data Aggregation," *Proc. 3rd Int'l Renewable and Sustainable Energy Conf.*, 2015, pp. 1–4.

[2.] G. Demiris et al., "Senior Residents Perceived Need of and Preferences for Smart Home Sensor Technologies," *Int'l J. Technology Assessment in Health Care*, vol. 24, no. 1, 2008, pp. 120–124.

[3.] H.-C. Chen et al., "Saliency-Aware Food Image Segmentation for Personal Dietary Assessment using a Wearable Computer," *Measurement Science and Technology*, vol. 26, no. 2, 2015, p. 025702.

[4]. V. Campmany et al., "GPU-Based Pedestrian Detection for Autonomous Driving," *Procedia Computer Science*, vol. 80, 2016, pp. 2377–2381.

[5.]. J.G. Andrews et al., "What Will 5G Be?" *IEEE J. Selected Areas in Comm.*, vol. 32,no. 6, 2014, pp. 1065–1082.

[6]. A. Abrol and R.K. Jha, "Power Optimization in 5G Networks: A Step Towards Green Communication," *IEEE Access*, vol. 4, 2016, pp. 1355–1374.