2018 4th International Conference for Convergence in Technology (I2CT)
SDMIT Ujire, Mangalore, India. Oct 27-28, 2018

AgrOne: An Agricultural Drone using Internet of Things, Data Analytics and Cloud Computing Features

Suhas M V[1*],Tejas S[1*],Snigdha[1*],Sitaram Yaji[2*],Sanket Salvi[3†]
*Department of Electronics and communication
†Department of Information Science and Engineering
*†Nitte Meenakshi Institute of Technology,
Bangalore, India
mesuhasmv@gmail.com[1],svyaji@yahoo.com[2],sanketsalvi.salvi@gmail.com[3]

***Abstract*—Ever increasing population and decreasing cultivable lands, has generated a pressing need of revising traditional agricultural practices. Especially, in country like India which was known as land of farmers, the problem of efficient and productive farming is severe at many levels. Modernization in farming practices is the only solution for this problem. This can be achieved by inventions in biotechnology, proper education and introduction of advanced tools in farming. One such tool could be Agricultural Drones. In this paper, we have discussed a Do-It-Yourself(DIY) approach to build own Agricultural Drone. Paper highlights, national and international work related to usage of modern computer guided tools, followed by our proposed design, components required, its assembly and implementation. We have used Cloud Computing and Internet of Things features for achieving this task. The Quadcopter autonomously fly and captures images of the field, which are processed to find the condition of the crops. This image and corresponding result is stored on Cloud. The On-Ground sensors and sensors present on the Quadcopter help to monitor the temperature and humidity for better yields. By leveraging usage of Cloud, user can have ubiquitous access to remotely monitored farm data. Results obtained shows, that using available technology how one can build Drone and On-Ground Sensor Devices for Agricultural Assistance. We have also discussed, future course of the project related to improvements in design and its potential extensions by performing advanced image processing and additional sensors.**

***Index Terms*—Agricultural Drone, Cloud Storage, Internet of Things**

I. INTRODUCTION

India ranks second worldwide in agricultural produce. Agriculture and its related domains like fisheries and forestry accounted for 13.7% of total GDP of our country in 2013 and about 50% of workforce. But, economic contribution of agriculture to Indias GDP is steadily declining due to increased urbanization and climatic changes. In an effort to prevent it from declining further, modern technologies are now replacing traditional methods of farming. One such modern technology is drones for agricultural assistance.

Initially, drones were considered only for military operations but, these days drones are getting popular in commercial, research and agricultural market space as well. Agriculture, is one of such sector which has high scope for drone assistance. Basic functions of agricultural drones includes remote monitoring, visual data collection, autonomous navigation and sensing weather parameters like temperature, humidity and light. Using this gathered data and some applied data analytics, we can propose necessary actions to be taken according to respective crop conditions. In addition to drones, usage of on ground sensors and actuators could result in collection of larger volume of data for efficient prediction. Generally, this large volume of data is stored on Cloud for Data Analytics and based on the results, respective actions can be taken. For example, Drones can be also used for spraying pesticides or to provide supplementary nutrients to crops.

In order to investigate and understand more closely problems associated with designing a modern age agricultural assistance, we propose a cloud based multi-modal data analytics framework for Smart Farming. Our proposed framework brings together the best of domains like Cloud Computing, Data Analytics, Internet of Things and Autonomous Drones. The remaining part of the paper is organized in following order. Section II, highlights some of the related projects done in this area. Section III, explains proposed framework and its related functional layers. Section IV focuses on hardware and software requirements, its usage and justification. Section V explains implementation proposed framework. Section VI shows results obtained at the end of implementation. Section VII gives conclusion and future work followed by references.

II. RELATED WORK

J. Wijitdechakul et. al.[1], had done multi-spectral image processing for determining health of crops and based on which corresponding actions are taken. However, computational resources required are high and implementation cost is more. Thus, for low cost solutions this system cannot be used though it is efficient. In our proposed framework, we have used concept of edge computing to offload processing from central server. Thus, a part of processing is done on Raspberry Pi mounted on Quad-copter and large data analytics is done and Cloud Service.

Paper[2], has helped us for understanding telemetry for autonomous drones and its related aerodynamics. The paper[2], highlights on usage of drones for spraying pesticides and its challenges with respect to changing aerodynamics due to dynamic payload. Although, our proposed work does not spray pesticides over the crops, this work has helped us to understand telemetry support for UAVs.

Paper[3], author proposes an architecture for drones equipped with appropriate integrating modules, sensors, cam-

978-1-5386-5232-9/18/$31.00 ©2018 IEEE

eras that can help for precise delivery of nutrients and pesticides/insecticides to crops. This has helped us for selecting drone and camera mounting location for good quality image capturing.

Manlio Bacco et al.[4], has provided a broad overview of challenges involved under different implementation scenarios. It also highlight the future of Smart Farming using interconnected farming vehicles. Paper[4], also discusses about robotic swarm for farming. Our proposed framework is inspired form this idea of collaborative sensing and centralized processing. Papers[5][6], uses basic sensors like temperature, humidity and soil moisture to determine field conditions. Using GPS location or device ID based on deployment positions, the location for sensed data can be determined. Based on values defined by respective algorithms corresponding actions will be taken by actuators. We have used this work to provide
additional data for supporting decision making.

III. PROPOSED FRAMEWORK

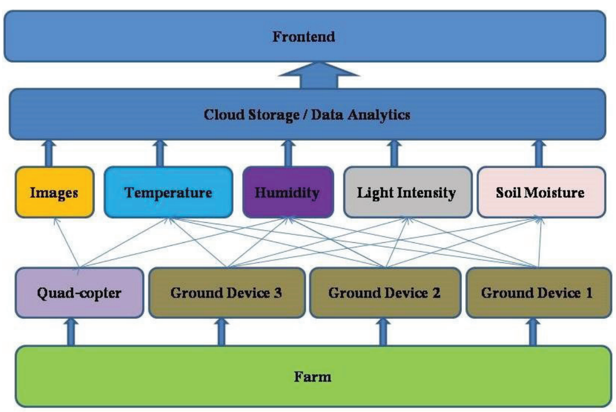

Fig. 1. Proposed Framework

The proposed framework contains following modules:

1) Autonomous Quad-copter: The Quad-copter should be able to carry payload of around 1Kg. This would enable us to mount Raspberry Pi and External Telemetry support devices. Raspberry Pi will be used for performing basic Image Processing on board. This will lead to improved resource utilization and it will also reduce load from the Storage Cloud. The external Telemetry Support will help us to make Quad-copter autonomous, since it uses precise GPS coordinates for navigation. Telemetry Support will not only navigate the Quad-copter but also maintain acceleration, pitch, yaw and altitude of the quad-copter.

2) On-Ground Sensor Nodes: We propose, a battery powered wireless Sensor Node consisting of Humidity Sensor, Soil Moisture Sensor, Temperature Sensor, and Light Intensity Sensor. Multiple such Sensor Nodes would be deployed in the field. Each Sensor node with Unique ID corresponding to its deployment location will be predefined. This will help to monitor and control different areas in the farm.

3) Image Processing: The Quad-copter will be mounted with an on-board camera. Based on input from the Telemetry System, once the Quad-copter reaches over a particular part of the field it will capture an image of it. This image will be processed by the Raspberry Pi connected to camera. The processed data will give information about the green component in the image, which we have used as indicator for crop health.

4) Cloud Storage/Data Analytics: Since, the computational and storage resources of Raspberry Pi are limited, we propose to use Cloud Services. In particular, Storage as a Service and Data Analytics as a Service. The images captured by Quad-copter will partially processed on Raspberry Pi itself, after which it will be compressed and uploaded to Cloud for Historical Data Log. Another cloud service is used for data visualization and logging.

5) Frontend: Frontend provides an interface for users to visualize data and understand it corresponding inference. It populates data by extracting it from both the cloud services. This can be hosted as a webpage or as mobile application.

IV. REQUIREMENTS

This section highlights the components needed and their usage for implementing our proposed framework.

As shown in Table 1, the sensor components are used for collecting data from corresponding environment. Sensors such as Humidity, Temperature and Light are common to both, Quad-copter and On-Ground Sensor Node. Whereas, Camera is exclusively for Quad-copter and Soil Moisture Sensor is exclusive for On-Ground Sensor Node.

Table 2, highlights different tools/languages/platforms used for implementing proposed framework.

V. IMPLEMENTATION

This section explains implementation of the proposed framework for Agrone. As shown in Fig. 2, we have three different layers in data life cycle. At first layer, we locally gather the data and optionally perform some simple processing depending on available computational resources. This layer is called Data Acquisition Layer. Next layer is the Data Storage/Analytics Layer. This layer comprises of the subscribed Cloud Services from respective Cloud Providers. The third layer is Frontend, which is used for consolidating data from different platforms and provide it as an unified interface.

The implementation of proposed framework will be explained in the form smaller interaction between each layers.

A. On-Ground Sensor Node - ThingSpeak

Fig. 3, shows connection between sensors and WeMos to form a Sensor Node. GND(Ground) of all sensors is connected to GND of WeMos. VCC(+5V) of WeMos is connected to VCC of DHT11(Temperature Humidity Sensor) and Soil Moisture Sensor. The signal pin of DHT11 is connected to D1(Digital Pin) of WeMos, signal pin of Soil Moisture Sensor is connected to A0(Analog Pin) of WeMos. The Anode of LDR(Light Intensity Sensor) is connected to D0(Digital Pin)

TABLE I HARDWARE REQUIREMENTS(A)

Sr. No.	Sensor		
	Component	Model	Justification
1	Humidity Sensor	SEN-10167	To monitor humidity of the
2	Temperature Sensor	SEN-10167	To monitor temperature of the room
3	Light Sensor	SEN-09088	To monitor ambient light in the room
4	Soil Moisture Sensor	SEN-13322	To monitor moisture content of garden soil
5	Camera	DEV-14028	To record and identify to objects
Sr. No.	**Microcontroller/Microprocessor**		
	Component	**Model**	**Justification**
1	WeMos D1 Mini	ESP8266	To forward the sensed data from Sensor Nodes to Cloud
2	Raspberry Pi	3B	To capture field images and to perform local basic image processing
3	GSM Module	Adafruit FONA SIM 800L	To send and receive data between Raspberry Pi and Cloud
4	Flight Controller	Ardupilot APM 2.6	To preset flight information such as altitude, path,speed, etc.
5	Quadcopter	REES52 F450	To take overhead images of the field

TABLE II. SOFTWARE REQUIREMENTS

Sr. No.	Software / Technologies	Usage
1	Python3	To implement Image processing, capture images and interact with cloud services
2	OpenCV2	To perform image processing
3	Arduino IDE	To write and upload code over WeMos
4	Heroku	To deploy our website
5	HTML+Javascript	To code for our webpage
6	Mission Planner	To design flight plan for Quadcopter
7	Cloudinary	To Store Images
8	ThingSpeak	To plot graphs based on received data

and Cathode is connected to GND of WeMos. WeMos is externally power by Batteries.

WeMos has on-board WiFi which is used to connect to nearest WiFi router. Once connected, this sensor node keeps sending sensed data from each pin to a ThingSpeak Channel. ThingSpeak provides APIs to enable this.

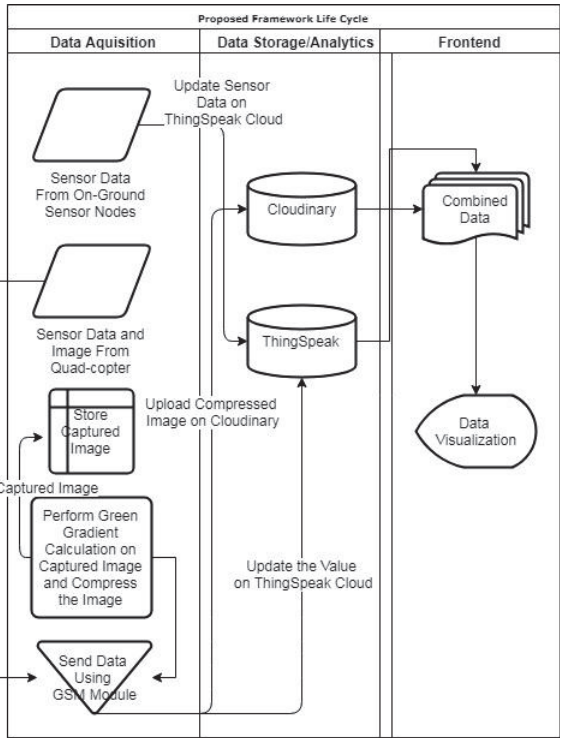

Fig. 2. Proposed Framework Life Cycle

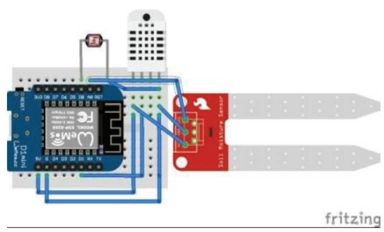

Fig. 3. On-Ground Sensor Node

To be able to send or receive data from ThingSpeak Cloud, we need to register first. Followed by creating own channel and data fields. Then by using unique API keys, we can update values over the channel. "ThingSpeak.setField(2,analogRead(A0));" this will read voltage value from Analog Pin A0 and set it to field 2 of ThingSpeak channel."ThingSpeak.writeFields(myChannelNumber, myWriteAPIKey);" this will write all the set values for respective fields and update it on channel using provided API key and Channel Number.

B. *Quadcopter - Cloudinary and ThingSpeak*

The quadcopter does not have prebuilt navigation control hence in order to fly it one has to be an expert. However, this cannot be expected in real life deployment of Agrone. Hence, we added a flight controller to automate the entire flight. We have used Ardupilot to achieve this. Fig. 4 shows the connection between Ardupilot and QuadCopter. Once this connection is done, we can upload a flight plan using Mission Planner Software.

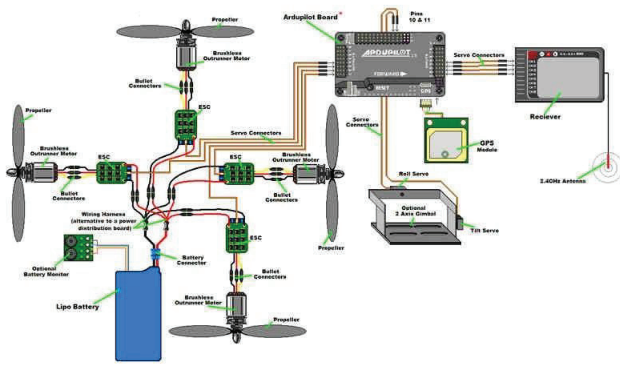

Fig. 4. Quadcopter+Ardupilot[7]

After flying the Quad-copter for several times on the same path, we calculated time required for reaching each location. We have predefined time at which images should be captured by Raspberry Pi. Following steps are followed on Raspberry Pi:

- From the captured image, we identify percentage of green pixels using standard hue saturation and thresholding methods provided by OpenCV. To achieve faster processing and better resource utilization as compared to single threaded program, we have used multithreading.
- The value obtained is then updated in ThingSpeak Cloud.
- The image on Raspberry Pi is compressed and uploaded to Cloudinary using provided APIs. Image is named as date time location.jpg by using simple string concatenation. This naming scheme also helps for maintaining sorted historical log.
- Apart from camera, Quad-copter also has DHT and LDR as shown in Fig. 5. DHT and LDR data is sent to ThingSpeak Cloud while image is uploaded to Cloudinary.

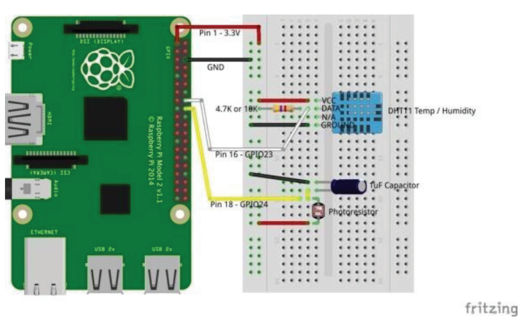

Fig. 5. Raspberry Pi+DHT+LDR

In order to send data over cloud using GSM module, we first need to interface Raspberry Pi with Adafruit FONA. Fig. 6, shows connection between Raspberry Pi and FONA.

Once the module is connected we perform module setup and testing as mentioned in [9]. The python code 'curl' command to upload data over ThingSpeak and Cloudinary.

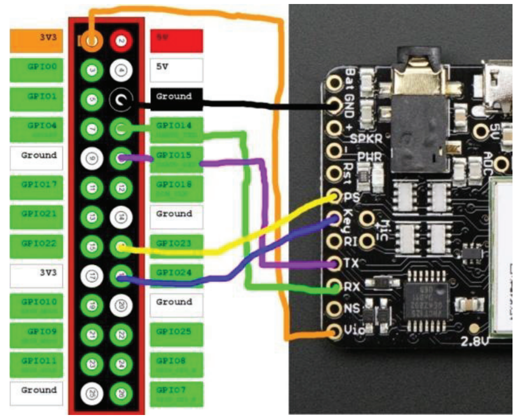

Fig. 6. Raspberry Pi+Adafruit FONA[8]

C. Cloudinary and ThingSpeak - Frontend

In order to provide a unified user interface, we built our own website and hosted it over free hosting platform named Heroku. The webpage basically loads the data from both the cloud service providers. We have used simple HTML and Javascript code to achieve this. Since its deployed globally, it gets ubiquitous access provided one has internet connection.

All these three interactions work in unison to provide required data with basic interface.

VI. Results

Fig. 7. ThingSpeak Interface

Fig. 8. Temperature and Humidity from Two Sensor Nodes

This section shows the obtained results. As project is more application based we have added screenshots of results. Fig.

7, shows ThingSpeak Cloud interface where we have logged into our channel feeds.

Fig. 9. Cloudinary Interface

Fig. 8 shows data plotted over different fields of channels. Left two graph in fig. 8, shows temperature and humidity from one sensor node, While right ones show details from other sensor node.

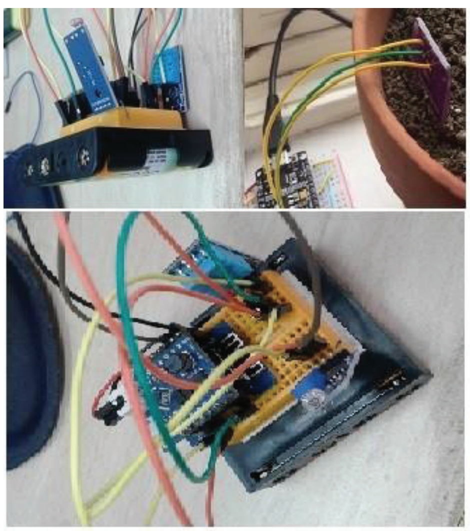

Fig. 10. Sensor node profiles

Fig. 9 shows the Cloudinary interface post log in. The heat map shows how many time particular image has been viewed. Fig. 10, shows the on-ground sensor node from different profiles.

Fig. 11. Webpage Index Page

Fig. 10,11,12, shows different interactions with designed frontend. Fig. 10, shows the index page of our website where it has two options. First one is to see health component of the field in terms of graph, also on clicking respective graph, the

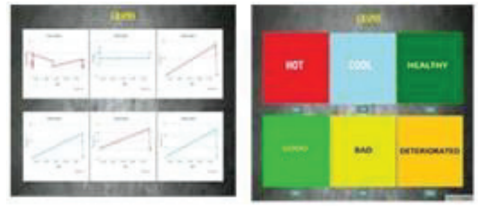

Fig. 12. ThingSpeak Graphs and Corresponding Inference

view toggles to one word inference of the graph. Second one is to see images taken on respective dates.

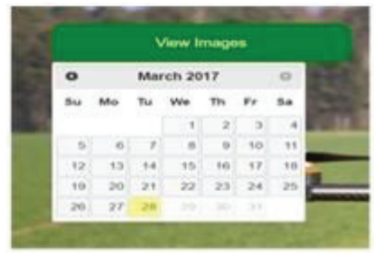

Fig. 13. Retrieving Image from Archive

Fig. 14. Overhead image of Farm

Fig. 15. Agrone:Agricultural Drone

Fig. 11, shows the graphs and corresponding inferences. Fig. 12, shows the selection of date for which images have to recovered. Fig 14. shows an image of farm field captured by quad-copter and Fig.15 shows the Agrone, our proposed quadcopter with Ardupilot Flight Controller, Raspberry Pi, Camera Module, GSM Module and Additional Battery Power supplied via Power Bank.

VII. CONCLUSION AND FUTURE WORK

We have implemented the proposed framework for provisioning agricultural assistance using Quad-copter. We have observed that camera stability and its exposure time affects the farm health calculation. We also found that, predefining time for capturing image works only if flight controller and raspberry are started simultaneously. However, due to lack of synchronization between raspberry pi and ardupilot the images captured were most of the times blur.

We have tested quad-copter and on-ground sensors separately in different environments, and both work as required. For, building better prediction algorithm data from both modules can be used with respect to time that it has been collected. We have built a very basic framework to provide proof of concept and which can be extended by adding more sensors and implementing advanced image processing techniques. The framework can further be improved by providing synchronization between On-Ground nodes and Quad-copter. By providing synchronization, power can be saved and On-ground Sensor nodes can use Quadcopter's GSM to upload data.

Further more, both the modules can be made solar powered for sustainable solution. The project can be further extended by adding actuators to perform tasks based on analyzed data.

REFERENCES

[1] J. Wijitdechakul, S. Sasaki, Y. Kiyoki and C. Koopipat, "UAV-based multispectral image analysis system with semantic computing for agricultural health conditions monitoring and real-time management," 2016 International Electronics Symposium (IES), Denpasar, 2016, pp. 459-464.

[2] D. Yallappa, M. Veerangouda, D. Maski, V. Palled and M. Bheemanna, "Development and evaluation of drone mounted sprayer for pesticide applications to crops," 2017 IEEE Global Humanitarian Technology Conference (GHTC), San Jose, CA, 2017, pp. 1-7.

[3] A. K. Saha et al., "IOT-based drone for improvement of crop quality in agricultural field," 2018 IEEE 8th Annual Computing and Communication Workshop and Conference (CCWC), Las Vegas, NV, 2018, pp. 612-615.

[4] M. Bacco et al., "Smart farming: Opportunities, challenges and technology enablers," 2018 IoT Vertical and Topical Summit on Agriculture - Tuscany (IOT Tuscany), Tuscany, Italy, 2018, pp. 1-6.

[5] S. Salvi et al., "Cloud based data analysis and monitoring of smart multi-level irrigation system using IoT," 2017 International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC), Palladam, 2017, pp. 752-757.

[6] R. F. Maia, I. Netto and A. L. H. Tran, "Precision agriculture using remote monitoring systems in Brazil," 2017 IEEE Global Humanitarian Technology Conference (GHTC), San Jose, CA, 2017, pp. 1-6.

[7] http://ardupilot.org/copter/docs/connecting-the-apm2.html

[8] https://learn.adafruit.com/network-interface-failover-using-fona/wiring

[9] https://github.com/initialstate/fona-raspberry-pi-3/wiki