Autonomous Quadrotor Control using Convolutional Neural Networks

Amer Hamadi [1], Dr. Abdulla Ismail [2]

[1]*Graduate Student, Dept. of Electrical Engineering, Rochester Institute of Technology, Dubai, UAE*
[2]*Professor, Dept. of Electrical Engineering, Rochester Institute of Technology, Dubai, UAE*

---***---

Abstract - *Quadrotors are considered nowadays one of the fastest growing technologies. It is entering all fields of life making them a powerful tool to serve humanity and help in developing a better life style. It is crucial to experiment all possible ways of controlling quadrotors, starting from classical methodologies to cutting edge modern technologies to serve their purpose. In most of the times quadrotors would have combination of several technologies on board. To control the quadrotor behavior for two different tasks, Obstacle Avoidance and Command by Hand Gesture, the use of Convolutional Neural Networks (CNN) was proposed, since this new technology had shown very impressive results in image recognition in recent years. A considerable amount of training images (datasets) were created for the two tasks. Training and testing of the CNN were performed for these datasets, and real time flight experiments were performed, using a ground station, an Arduino microcontroller and an interface circuit connected to the quadrotor. Results of the experiments, at the end of the paper, show an excellent accuracy rates for both tasks.*

Key Words: **Quadrotor, CNN, Convolution, ReLU, Softmax, Microcontroller.**

1. INTRODUCTION

UAV (Unmanned Aerial Vehicle) is an aircraft without a pilot, depending mainly on autonomous or remote flight control, this system is used in recent years in many civil and military applications, providing many advantages over manned systems such as reduced cost, no risk on crew for hazardous missions, maneuverability and long endurance [1]. UAVs can be classified according to size , range, altitude or number of rotors , Table 1 shows the possible classification of UAV.

Table -1: UAV Classification

UAV				
Size	Range	Altitude	Wing Config.	No. of Rotors
Micro (MAV)	Close range	High Altitude Long Endurance HALE	Fixed Wing	Single Rotor
Mini (MUAV)	Short range	Medium Altitude Long Endurance MALE	Flapping wing	Multi rotors
Nano (NAV)	Medium Range Endurance		Blimps	

QUAV (Quadrotor UAV) is a Multi-rotors UAV that is lifted and propelled by four rotors. It is considered a benchmark research platform because of its high manoeuvrability and simple mechanical structure. However the control design for this type of vehicles is a complex task.

2. Convolutional Neural Network

Convolutional Neural Networks (CNN) learn a complex representation of visual data by exposing to vast amounts of data, they are inspired by human visual system and learn multiple layers of transformations, which are applied on top of each other to extract progressively more sophisticated representation of the input.

Initial layers capture low level features such as edges and corners, then followed by middle layers capture mid-level features like object parts, and the last layer captures high level-class specific feature such as object model and face detector as shown in Figure 1 [2].

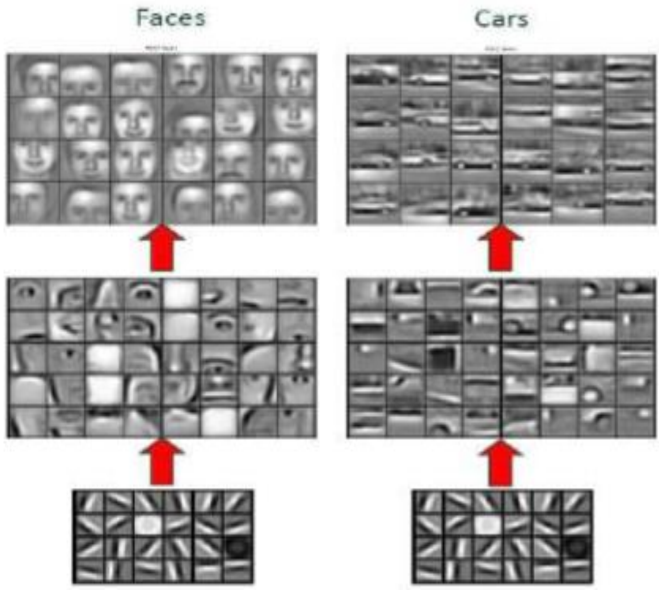
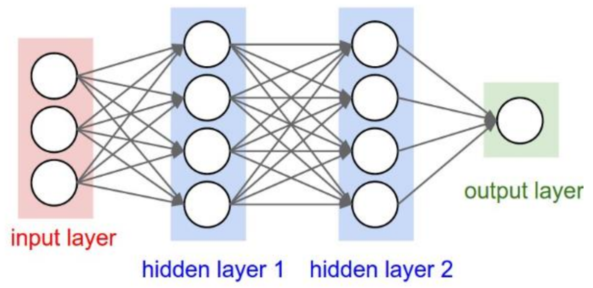


Fig-1: Convolutional Neural Network learning representation

A classic Convolutional Neural Network consists of a multiple convolutional and fully connected layers, in which most of the operations are executed; pooling layers that are used to evade over-fitting; a classification layer and to classify final results into classes. Each layer in the CNN comprises of 3D volumes of neurons, with width, height, and depth as shown in Figure 2 [3].



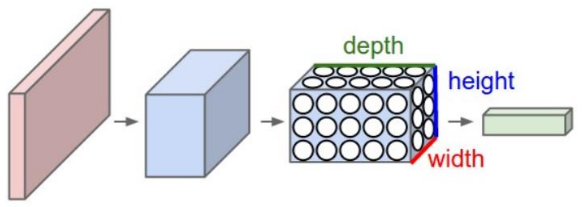


Fig -2: Up : Classic 4 layer Neural Network. Down: A CNN layer that arrange its neurons in three dimensions (width, height, depth).

3. CNN Implementation

In the last six years deep learning has massively altered the domain of machine learning, computer vision, pattern recognition, robotics etc. It has been proved that deep learning is capable of achieving better detection results than traditional techniques, because of its filtering through its multiple layers [4]. In this paper two tasks were selected using the same setup:

• Obstacle avoidance using a 15 layers and 32x32x3 pixel images.

• Command by gesture which uses Transfer Deep Learning from AlexNet, with 25 layers and 227x227x3 images.

3.1 Obstacle Avoidance

A dataset is made with 4 classes, each class has 130 images of the obstacle position inside the image frame (right , left, up, down), Figure 3 shows sample of two class shots (left and right) for this project.

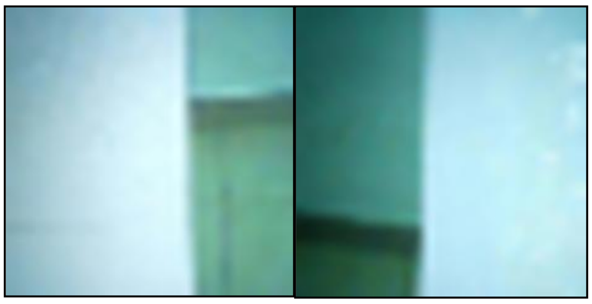


Fig -3: Two sample pictures for Left and Right classes each is 32x32 pixels

The first convolutional layer were chosen to have to dimension of 32x32x3 for the sake of computational efficiency, therefore the input images were resized accordingly, the training and validation images were chosen randomly 80% and 20% of the total images respectively.

The second layer will be pooling layer to decrease the dimensions the following are alternating with pooling. There are three convolutional layers on the second fifth and ninth layers and the last layer is Softmax layer for classification. The number of filters used in the pooling layers and the dimensions of each layer is detailed in Table 2.

Table -2: Obstacle Avoidance CNN Layers

#	Layer name	Description
1	'imageinput'	Image Input
2	'conv_1'	Convolution
3	'maxpool'	Max Pooling
4	'relu_1'	ReLU
5	'conv_2'	Convolution
6	'relu_2'	ReLU
7	'avgpool_1'	Average Pooling
8	'conv_3'	Convolution
9	'relu_3'	ReLU
10	'avgpool_2'	Average
11	'fc_1'	Fully Connected
12	'relu_4'	ReLU
13	'fc_2'	Fully Connected
14	'softmax'	Softmax
15	'classoutput'	Classification Output

Results:

The CNN network was trained with hundreds of obstacles photos with different orientation, the success rate is measured with a matrix known as Confusion Matrix which is shown in Table 3, are obtained from training and validation phases. The training required 20.83 seconds on the GPU. The mean accuracy rate was 75%.

Table -3: The Confusion Matrix (ConfMat)

	Up	Down	Left	Right
Up	0.6364	0.3636	0	0
Down	0	0.7714	0.1429	0.0857
Left	0	0.1351	0.8198	0.0541
Right	0	0.175	0.05	0.775

3.2 Command by Hand Gesture

The second CNN experiment implemented in this paper uses Transfer Learning. The main drawback of CNNs is the requirement of vast training datasets which need long computational time and special hardware during training. Yet, testing time is very less which qualify it to meet the requirement of real time applications. To defy the requirement for large datasets, a concept called Transfer Learning is usually used.

Transfer learning is the technique of using a pre-trained model with the weights and parameters of a network that has been trained on a large (AlexNet in this paper) and since the images are somewhat similar in nature a "fine-tuning" is done for the model with the new dataset. The fine tuning is done by replacing only the last layers with the new classifier and keeps all of the other pre-trained layers which will act as feature extractor, and then re-train the network normally. In our case the 23rd layer is replaced because AlexNet has 1000 neuron in it to classify 1000 objects, while the requirement here is only five gesture images for UP, Down, Left, Right and Stop. Figure 4 shows two examples of the gesture control images used in this experiment.

Fig -4: Left and Right gesture command images added to the dataset

The 25th layer also will be replaced with a new layer that classifies the five gestures above, the layers comparison before and after modification shown in Table 4.

Table -4: A comparison before and after modification of AlexNet layers

#	Name	AlexNet	Modified
1	'data'	Image Input	Image Input
2	'conv1'	Convolution	Convolution
3	'relu1'	ReLU	ReLU
4	'norm1'	Cross channel	Cross channel
5	'pool1'	Max Pooling	Max Pooling
6	'conv2'	Convolution	Convolution
7	'relu2'	ReLU	ReLU
8	'norm2'	Cross channel	Cross channel
9	'pool2'	Max Pooling	Max Pooling
10	'conv3'	Convolution	Convolution
11	'relu3'	ReLU	ReLU
12	'conv4'	Convolution	Convolution
13	'relu4'	ReLU	ReLU
14	'conv5'	Convolution	Convolution
15	'relu5'	ReLU	ReLU
16	'pool5'	Max Pooling	Max Pooling
17	'fc6'	Fully Connected	Fully Connected
18	'relu6'	ReLU	ReLU
19	'drop6'	Dropout 50%	Dropout 50%
20	'fc7'	Fully Connected	Fully Connected
21	'relu7'	ReLU	ReLU
22	'drop7'	Dropout 50%	Dropout 50%
23	'fc8'	Fully Connected layer (1000 neurons)	Fully Connected layer (5 neurons)
24	'prob'	Softmax	Softmax
25	'output'	Classification Output 1000 classes	Classification Output 5 classes

Results:

The training using different gestures photos required 180.36 seconds on the GPU. The results obtained from training and testing phases is with mean accuracy rate equals to 98%. Which significantly greater than the 15 layer network used in the first task, and it is proof of the robustness and flexibility of AlexNet to adapt new tasks other than it was initially designed for.

The first convolutional layer 96 weights are 11x11x3 filters can be displayed, from which we can notice that they are representing a useful information that can describe for example a horizontal edge or a blob, each filter will have a unique feature that will pass through the examined input picture to detect that feature as mentioned in Section 2, Figure 5 shows the a montage of first convolutional layer for this network.

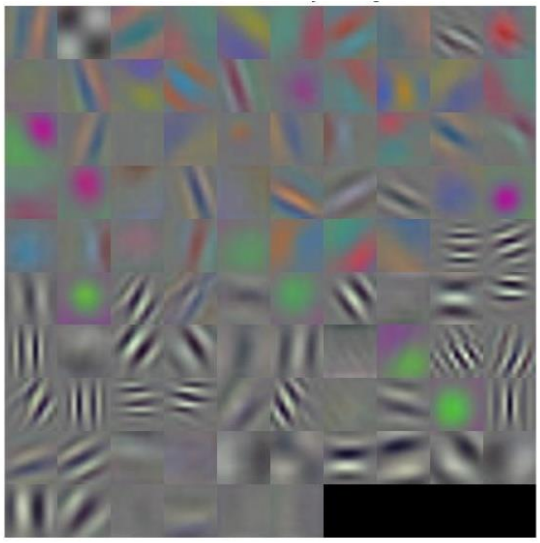


Fig -5: Filters of the first convolutional layer

4. Test-bed Setup

The setup is designed to have the inner loop responsible for the attitude control and typically uses PID controller located in the quadrotor, and the outer loop is responsible of the position performed by the ground station as shown in Figure 6.

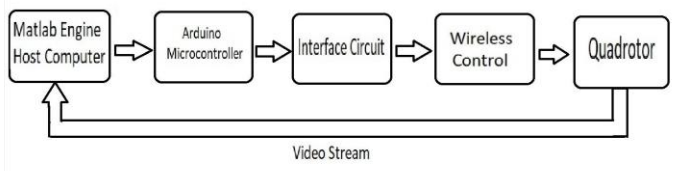


Fig -6: System block diagram ample paragraph.

The ground station is a computer with processor Intel i7-4150U @2.60 GHz , 8GB memory and GPU NVIDIA Geforce 840M, dedicated 4GB memory with capability to work with CUDA library required for the CNN computation. MATLAB codes are done for two different tasks, communicating via a serial port RS232 with an Arduino microcontroller which is sending commands to the quadrotor as shown in Figure 5.6 , the feedback loop is the real time video stream sent by the quadrotor onboard camera.

The Arduino ports used to control the attitude and altitude of the quadrotor is shown in Table 5.

Table -5: Arduino ports assignment

Arduino Port	Control Action
D6	Roll Right
D11	Roll Left
D8	Hover Up
D9	Hover Down
D10	Pitch Forward
D7	Pitch Backword

An interface circuit was designed to connect the Arduino ports to the Wireless controller, mainly using photo coupler as shown in Figure 7, to eliminate any noise interference back to Arduino and host computer.

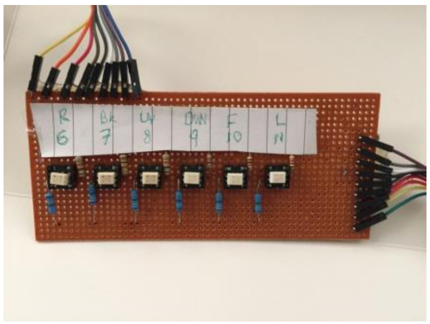


Fig -7: The Interface Circuit

4.1 Programming Quadrotor Behavior

For both tasks the quadrotor will be starting automatically to a certain elevation, then start sending the video stream to the host computer, for Obstacle Avoidance system the CNN will recognize the obstacle position in the captured image frame then according to the classification MATLAB code will be sending commands to the quadrotor to avoid the obstacle, if the obstacle recognized at left the quadrotor will be rolling to right then pitching forward then rolling Left again as shown in Figure 8.

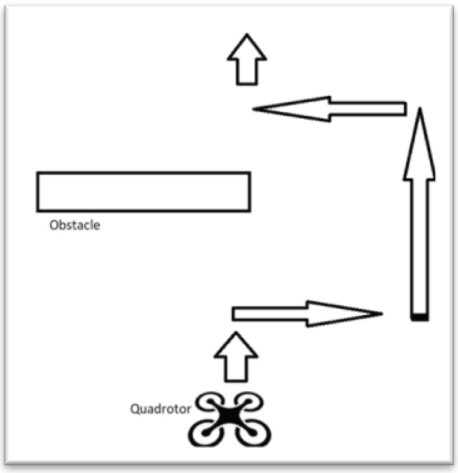


Fig -8: CNN classification of obstacle orientation and Quadrotor reaction behavior

In the Command by Hand Gesture task the quadrotor will be elevated to a certain height then send a photo capture to the CNN on the host computer which will recognize the gesture as left or right and send the roll left or right command accordingly as shown in Figure 9.

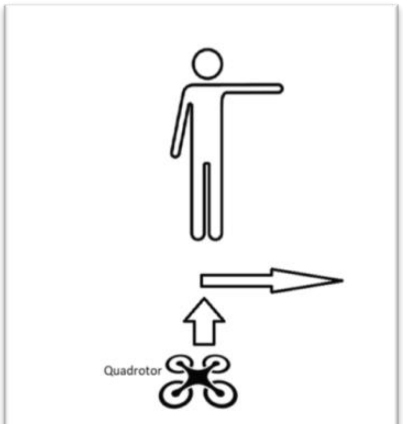


Fig -9: CNN network gesture recognition and Quadrotor response

4.2 Real-Time QUAV Flight Experiments

The QUAV system was built and test flights were implemented for both tasks. The Quadrotor was driven by the ground station as explained earlier in the Figure 6; the video for both experiments with brief details is given in the link https://youtu.be/EslaUyijXZo .

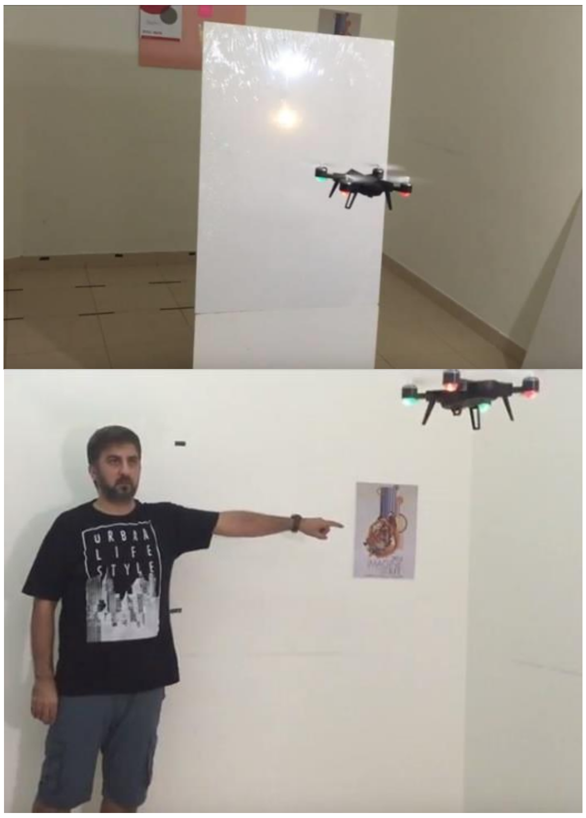


Fig -10: Video screenshots for the system implementation

In Part A of the video the CNN is pre-trained for the Obstacle Avoidance task, the program sends a command via Arduino to the Quadrotor to hover up to a certain elevation, then the on-board camera captures the obstacle in front of it and sends it back to the CNN which classifies it as right, left, up or down. The program accordingly decides the behavior required as explained in Section 4.1 and sends a sequence of movement commands to the Quadrotor to avoid this obstacle.

In Part B, CNN is pre-trained for gesture recognition task. The Quadrotor camera sends a photo of the operator in front of it back to CNN which classifies the operator's gesture according to the training; the classification result will pass to the code of program which decides consequently the behavior commands need to be sent to the Quadrotor through wireless controller.

5. CONCLUSION

In this paper two CNN structures were selected to implement two tasks; the first one used a 15 layer network to detect obstacles in front of a quadrotor. The results were acceptable although there were limited number of dataset images. The second task was to use a transfer deep learning technique from a pre-trained AlexNet to recognize an operator's gesture, which resulted in an excellent accuracy rate. Both tasks were implemented on a real quadrotor using host computer, Arduino microcontroller and an interface network to control the quadrotor, the experiment was successful. The system performance reflects a major advantage of scalability for classification of new classes and other complex tasks, towards an autonomous flying and more intelligent behavior of quadrotors.

REFERENCES

[1] Samir BOUABDALLAH et al.," Design and Control of an Indoor Micro Quadrotor "2004
[2] Charles Siegel, Jeff Daily, Abhinav Vishnu, "Adaptive Neuron Apoptosis for Accelerating Deep Learning on Large Scale Systems".
[3] Andrej Karpathy, Fei-Fei Li "CS231n Convolutional Neural Networks for Visual Recognition".
[4] Z. Chen, O. Lam, A. Jacobson, and M. Milford, Convolutional Neural Network based Place Recognition," 2013.

BIOGRAPHIES

Amer Mahdy Hamadi is a graduate of Electronics and Communications Engineering from Al Nahrain University Baghdad, Iraq. Currently pursuing his Master's degree in Electrical Engineering, specializing in Control System, at Rochester Institute of Technology (RIT), Dubai Campus.

Dr Abdulla Ismail obtained his B.Sc. ('80), M.Sc. ('83), and Ph.D. ('86) degrees, all in electrical engineering, from the University of Arizona, U.S.A. Currently, he is a full professor of Electrical Engineering and assistant to the President at the Rochester Institute of Technology, Dubai, UAE