Autonomous Navigation of a Quadrotor in an Indoor Environment using Deep Learning

Shanmuga Perumal Harikumar,Master of Science in Robotics Engineering,WPI

***Abstract*—The purpose of this project is to develop an autonomous quadrotor system that could navigate an indoor environment without running into obstacles. This documents outlines the project in which the author wishes to train a deep neural network to generate motion commands based on the input image. The sensor data is directly mapped to the control input to the system by the deep learning network. An initial problem statement is formulated. The final project goals are enumerated. Previous works in the field relating to the project undertaken are explained in brief. The various hardware and software components to be used in the project are listed,along with a short description of each. Finally the proposed project logistics is outlined.**

***Index Terms*—Deep learning, Autonomous Navigation, Convolutional Neural Networks, Quadrotor Indoor Navigation**

I. Introduction

The purpose of this project is to make a quadrotor learn navigate through an indoor environment autonomously using deep learning. In other words, the author hopes to map sensor input directly to motion commands.The quadrotor must learn to detect and avoid obstacles in its path. The application of such a learning system is immense. The quadrotor market is currently largely focused on outdoor quadrotors that rely heavily on GPS for positioning and way-points navigation. There is however a dire need for quadrotors that can operate in an indoor environment. There are several problems that need to be addressed in order for a quadrotor to successfully fly indoors. The problem of navigating through a cluttered indoor space such as a warehouse or an office is yet to be solved. This project aims at using deep learning to solve this problem.

The majority of current research in this field of indoor navigation relies heavily on lidars and creating a 3D map of the environment. Creating a 3D in itself is both computationally expensive and memory intensive. SLAM algorithms require a huge amount of computing power to build the maps and localize based on the maps. Navigating such a mapped environment is a whole other topic. Motion planning involves collision checking which takes the majority of the planning time and is again computationally expensive.

Unlike the above mentioned methods that use features that are handcrafted by humans in order to localize and map a place, Deep learning networks can do all this implicitly. Deep learning networks have been proven time and again that they are better at tasks such as object recognition, classification and segmentation than the other methods. Such networks with minor tweaking can be used to give commands to UAVs based on visual input data. The process of controlling a MAV can be modelled as a classification problrm that the DNNs are really good at.

Convolutional Neural Networks(CNN) have been used in object recognition with high success rates.Before CNNs object recognition was done by using hand crafted features which required domain experts and required special features to be developed for each specific application which was laborious. The development of Deep Neural Networks(DNN) led to the process of classification becoming more general.

A. Problem Statement

In this project the author hopes to train a deep neural network to detect obstacles and navigate in an indoor environment avoiding the detected obstacles. INitially the plan was to use as inputs to the network image and depth information from a RGBD camera(Intel Real Sense) along with IMU data as labels. Since the Real sense camera was rendered useless due to an accident, a parrot AR Drone with a monocular camera was used and the images from the camera along with the commands will be given as inputs to the drone.

II. Previous Work

There have been several previous attempts at using Deep Neural Networks for navigation of a robot.[1] used DNN to navigate a forest trail autonomously. The training data was obtained by using three cameras mounted on the helmet of a hiker. One of the cameras was pointing straight,one towards the left and one towards the right. The hiker was made to follow the trail while always turning towards the direction the trail was heading in.The cameras pointing towards the left always had the trail towards its right and the camera pointing towards right always had the trail to its left.The images were also flipped and the dataset was augmented.The problem was treated as a classification problem. The final network took an image as a input and gave the direction towards which it thought the trail was in as output. The final output was something like a probability. The difference between the probability of turn left and turn right commands was used as the yaw command and the probability of going straight was used as the velocity.

[4] uses a somewhat similar approach to control a car using an input image. The input from the sensor is mapped to control commands for the vehicles to follow. The data acqisition setup has three cameras ,one pointing straight and the other two at a fixed angular offset on either sides. The dataset was further augmented by distorting the input images by applying fixed transformations. The data was obtained from various roads

Fig. 1. Parrot AR Drone

under various weather conditions,to ensure good learning and generality. The learned parameters were first tested on the simulator and subsequently on a real car.The authors claim that the car was autonomous 98% of the time. This shows that it is feasible to map the sensor inputs directly to control outputs

[2] uses the depth information along with the rgb data for the process of object detection. the rgb images and the depth information are processed as different channels. Pretarined model as are first used an then the separate channels are trained separately and finally the networks are combined and is trained one final time.. one more novel approach is hoe they do scaling of the images ,instead of just warping the image , the authors use tiling the borders along the axis of the shorter side. Since the depth data also contains features such as edges and corners that are visible in rgb images the depth channel was also trained using the image net database. This paper also talks about the various methods of encoding the depth data namely 1)rendering the depth data as greyscale images and then replicating across three channels, 2)using surface normals 3)HHA encoding,where the three channels are height above the ground ,horizontal disparity and pixelwise angle between surface normal and gravity direction.

[3] explored the depth parameter of the neural network to improve the accuracy. They experimented with a variety of network architectures. VGG 19 and VGG 16 two of the most successful networks according to the experiments conducted by the authors ,performed exceptionally well at classifying and also a variety of other tasks

III. Hardware and Software Used

A. Parrot AR Drone

Parrot AR Drone is a cheap MAV that is well suited for research purposes. The drone has a forward facing camera,IMU,downward facing camera for optical flow,a downward facing ultrasonic sensor for altitude estimation. The drone flies stably iut of the box,with optical flow running internally.

B. Robot Operating System(ROS)

Robot Operating System(ROS) is a software framework for development of software for robots specifically. It is an opensource implementation and has a very active community.

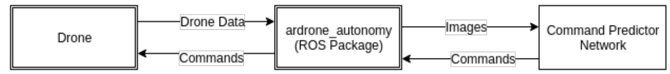

Fig. 2. System Overview

Using ROS simplifies communication between the various parts of the system and help visualize the entire system better. There are hundreds of packages available in ROS ranging from perception to motor drivers.

C. Parrot ROS Package

Parrot provides a SDK that lets programmers create their own applications for the drone. There is a ROS package available that transmits data from the drone in separate topics and also sends data back to the drone in topics. This makes development much easier for a roboticist. This also makes the drone readily interfaceable with other open source ROS packages available.

IV. Body

A. System Overview

The drone communicates with the Neural network through the ardrone_autonomy ROS package. The package publishes all the relevant information about the drone in separate ROS topics and the network input is the image from the front facing camera. The resulting commands is sent as Twist commands to the drone over another ROS topic through the ardrone_driver node.

B. Data Collection

For the purpose of this project the indoor environment chosen was warehouses.This was perfect for a deep learning navigation network because of the aisles and regularly arranged boxes. This permits for the DNN to learn the necessary features to track and follow. This also avoids the drone from getting close to the shelves and avoid collisions. Initial idea was to give the IMU data directly as input, but this was discarded as the IMU data was too noisy and the final system trained using such noisy data would not be robust.The drone was flown along multiple aisles and commands were recorded along with the images from the front facing camera. Human pilot gave the drone instructions to move forward,left right or yaw left, right. This will be the output of the classifier as well.

C. Networks

D. Training

E. Analysing the output of the layers

V. Experiments

VI. Conclusion

VII. Acknowledgements

I would like to thank professor Carlos Morato for his guidance and invaluable inputs.The assignments and paper reviews were instrumental in helping me understand the concepts of deep learning.I would like to thank the CEO of 89Robotics,

Fig. 3. Parrot in warehouse

Mr.Jackie Wu for permitting me to use the company property for the purpose of the class project and also for driving me to the warehouses and helping in the collection of data. I would also like to thank the CTO Mr.Spencer Williams for being patient with me and working around my schedule for the Deep learning course to assign me tasks.

REFERENCES

[1] Alessandro Giusti,Jerome Guzzi,Dan C. Cirezan,Fang Lin HE, *A Machine learning Approach to Visual Perception of Forest Trails for Mobile Robots*, IEEE Robotics and Automation Letters,Nov 2016

[2] Andreas Eitel, Jost Tobias Springfield,Luciano Spinello, Martin Reinmiller, Wolfram Burgard,*Multomodal Deep Learning for Robust RGB-D Object Recognition*

[3] Karen Simonyan , Andrew Zisserman, *Very Deep Convolutional Networks For Large-Scale Image Recognition*

[4] Mariusz Bojarski,David Del Testa, Daniel Dworakowski Bernhard Firner, *End to End Learning for Self Driving Cars*, Nvidia Corporation

[5] DJI website, *www.DJI.com*,DJI 2016

[6] Deep learning, *http://deeplearning.net/software/theano/*

[7] Intel real sense, *https://software.intel.com/en-us/realsense/devkit*