Autonomous Vehicle Processor

PROJECT PLAN

Dec1710 - Team VOLDEMORT
Rockwell Collins - Josh Bertram
Advisors - Philip Jones, Joseph Zambreno
Team Leader - Alex Orman
Communications Leader - Sean Jellison
Webmaster - Chris Kelley
Key Concept Holders - Lixing Lin, Evan Lambert
Scribe - Lucas Ince
dec1710@iastate.edu
http://dec1710.sd.ece.iastate.edu/

Revised: 04/20/2017, Ver. 3.0

Contents

1 Introduction

1.1 Project statement

The purpose of this project is to utilize an onboard system in realtime to detect the location of objects in a drone camera's view. To accomplish this we use a neural network to identify complicated objects and then a series of other, simpler methods to extract additional information such as distance away from the drone.

1.2 Purpose

This software will utilize a system's GPU to be able to more efficiently and quickly apply deep learning to embedded real time system processing. It will be able to detect and analyze objects and provide relevant information about it.

The use of a neural network with real time data on on-board hardware has a wide range of applications beyond this project. While many military applications come to mind, there are plenty of civilian uses as well to better the world. A good example of this would something like Amazon's drone delivery service. If they could utilize this system, it could save time and money for the company by avoiding objects and ensure more happy customers by putting packages in more convenient and secure locations.

1.3 Goals

When finished, the system will be able to be installed onto any autonomous system and be able to provide accurate and useful information about the surroundings. This is intended for use on remote controlled drones, but this could also be applied to most autonomous vehicles to detect objects near the vehicle or surveillance systems. The ability to detect a generic object has a multitude of uses already, but our system will be able to be trained on specific relevant objects. Additionally, the ability to easily detect novel objects after training.

Lastly, here is a list of tentative features we hope to achieve in order of under taking:

- Simulation data collection
- Distance and relative object orientation calculation
- Geographical position detection based on landmarks
- Object trajectory predictions
- Multi Camera and alternative camera support
- Wingmate and other aerial object tracking

2 Deliverables

The aim of this project is to provide software that can be installed onto an embedded board for use in autonomous aircraft. The software will be able to detect landing strips, runways and known airports. A stretch goal being able to analyze additional flying objects, be able to track objects and point out landmarks or significant structures.

Specifically, these items will be released to the client upon completion:

1. Source code for application developed, unless otherwise specified we(the students) may share and reuse any and all source code.
2. A PDF guide explaining installation and usage of source code aforementioned
3. A powerpoint explaining the project
4. Any videos generated during this project regarding the project.

3 Design

To give a high level description, this project will utilize a TK1 board mounted on a aerial platform(AKA drone) to analyze images captured from an onboard camera. This system will then feed the information of what it sees into other onboard systems.

The TK1 has been selected due to the lower cost and ease of use with our selected learning framework. The TK1 is also tailored to machine learning tasks since most tasks manipulate large matrices. The Jetson TK1 has the same advanced features and architecture of a modern desktop GPU while still using the low power draw of a mobile chip.

We chose TensorFlow because we have the most experience with it and because Tensorflow has a large support community and extensions such as Keras to simplify coding. Additionally, Google has released a pre-trained neural network that we would like to utilize in our design. This pretraining can save us a good amount of time and increase our accuracy.

The aerial platform, in this case likely a drone, will be used but will be provided by the client on as needed and approved basis. This will not be an issue until later stages of the project.

It is important to specify that the drone will be part of the project as an aerial vehicle as drones can have a large amount of disturbances while taking pictures from a mounted position. This means we must carefully choose our camera and training system as stills taken from the internet or video captured via other methods may not be what the end results sees.

3.1 Previous work/literature

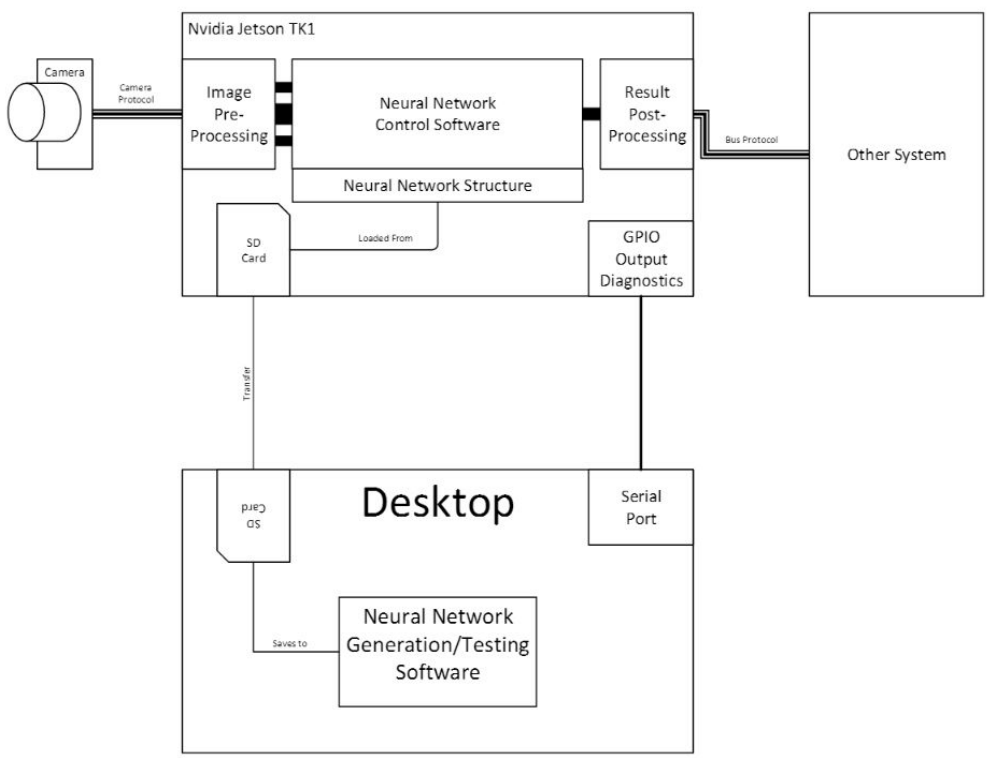
1) One similar project[1] to ours, created by a university research group in Austria, highlighted the plausibility of this project working. In their project they built a drone to be controlled by a neural network that took in live data and transmitted it back to a ground

station. The ground station, in this case a laptop, computed the neural network and then transmitted instructions back to the drone.

   a) https://www.youtube.com/watch?v=umRdt3zGgpU

2) Matt Havey posts in his blog [2] about how we went about using a raspberry pi and a camera to do on the fly analytics like we are. In his case, he wanted to know if football was on the TV or if was commercial. Many of the method he discusses in the post gave us a good starting point and more importantly validated that this project was actually a possible.
3) Havey continues on in another post[3] where he explains different ways we can utilize machine learning and machine vision, and the associated trade offs. He also talks about utilizing a different method on his Pi experiment in hopes of decreasing processing time, which we are interested in.
4) Pilot AI Labs, Inc. is a startup founded by Stanford students are working on a deep-learning based computer vision platform to solve real world problems on computer-constrained embedded devices. They currently have a drone that is capable of following someone around. This project is similar to ours in the sense that it uses deep-learning, an embedded system, and does some calculations based on what the neural network predicts.

3.2 Proposed System Block diagram



(System Diagram 1)

3.3 Assessment of Proposed methods

- **Computer vision with machine learning.**
  - This method would combine standard image processing techniques together with a basic neural network or other machine learning method.
- **Deep neural network.**
  - This method would involve using a neural network directly on the image data. This network would be larger than the other methods.

3.4 Validation

For our project we have several steps in which needs verification. Each step thus must have a unique verification process.

- Model Training
  - To start, we need to ensure that our model is progressing in the correct direction, that is to say not losing the ability to detect things correctly via overfitting. To help mitigate this issue we will be carefully controlling the data sets we feed it for training as well as verifying select results by hand to ensure automated methods are producing meaningful results.
- Software and Hardware Compatibility
  - The drone we will be equipped to is mostly a black box to us. So for this reason we need to ensure that we carefully follow any specifications given to us by our host devices on the drone. Ultimately the verification for this will come our client.
  - Furthermore, we need to ensure that all our components(ie camera, board, and software packages) all work together. We have spent extensive time researching these issues and believe them to not be a problem.
- Model Export and Embedded accuracy
  - This stage is the most uncertain. We can verify that the model exports correctly but we also need to ensure that the model operates correctly in a different system and can properly run on our onboard system. To ensure accuracy at this step, we will simply run our diagnostic tests from the training stages on the board itself.

4 Project Requirements/Specifications

4.1 Functional

- Runable on the embedded system.
- Can detect basic objects.
- Must be able to take in images from a camera
- Be able to locate key features in an image

- Be able to identify known objects or features from 400ft - 600ft

4.2 Non-functional

- Embedded system must fit inside the drone
- Must be able to process an image at least once per second from onboard camera
    - Ideally, target goal is 30-60 FPS

5 Challenges

5.1 Expected Challenges and Difficulties

Our first concern is with hardware compatibility. We are planning on using an Nvidia Jetson board and would need to incorporate the board with a drone. The drone has a tight constraint on space for additional peripherals. The board may need to be modified to fit in the space provided, however Rockwell Collins has been experimenting with this. Since we are dealing with training neural networks, these neural networks can become quite complex with more and more features. Being able to fit the neural network on the Jetson board and being able to gather enough information in a reasonable amount of time to make an accurate prediction is a real concern. Finally, the objects we are trying to detect will be at long distances, and accuracy is dependent on the camera and the network. Issues where an object is detected at close range may not be visible at longer range. The reverse is also true. Similarly weather and lighting conditions can affect quality of the pictures taken, but we plan to only be operating in clear weather.

5.2 Risk Assessment

This project is going to be run on an embedded board inside a flying drone. This is largely a research project and neural networks require some powerful hardware to run. It is therefore possible that the system cannot work on the board we use.

Hardware failures are a possibility, which would cause time delays at best, and require replacements or alternatives at worst. We expect this project to push our board to its limits so overheating is our main concern, but hardware incompatibility can can failure as well.

We will be using some software and libraries that are not owned by us. Anything we use will be under general public use license.

6 Timeline

6.1 First Semester

The first semester will be used to first learn about neural networks and be able to understand how to implement them in TensorFlow. Once we have a strong idea of how neural networks function, we'll determine how to break down the project to implement the features we want and create a plan to implement them. There is some concern about TensorFlow not being able to run on the TX1 board we specified. If this becomes an issue later on, we are open to adapting to a new library, such as Caffe.

By the end of the semester, we want to have basic functionality implemented, such as camera control and image input, simple image processing, and image analysis. This should all be able to be run on a personal computer, but in such a way as to make it easy to implement on an embedded board. The system should also be able to be trained for particular objects. This is one of the more important parts for the system as this is what allows the network to determine if a particular object is visible.

Additional features we want the system to be able to do are not expected to be started until next semester, however with the number of people we have working on this, we should be able to have extra features at least started.

We'll be designing a user interface as well to display information we collect. The UI would be used for training data output as well as providing important information back from the board such as type of object, orientation of object, and distance calculation to the object identified.

We will also need to collect images to analyze. One of the hardest parts of any neural network is finding large data sets to put into it. To get the images we need, we will be using a free flight simulator to get images of aircraft for aerial detection, and we will either use google earth, or a drone with a camera to take many pictures of objects we could test. We will be collecting images throughout the semester, and possibly throughout next semester, for different objects.

March

- Basic neural network setup
- Hello World on desktop

April

- Basic object detection
- Camera support implemented
- Simple image processing

May

- Basic functionality implemented and working
- Object recognition working on computer
- Embedded system hello world

The last thing we want working at the start of the first semester, is a functional "Hello world" equivalent for neural networks working on the embedded board.

6.2 Second Semester

Since we will have two boards available, the Jetson Tk1 and Jetson Tx1, we want it to be able to run on the Tk1. We want to be able to determine early on if the Tk1 is going to be powerful enough, or if we must use the Tx1. We expect implementation onto the Tx1 to be more difficult due to some discussions we've read online on the board.

Some features we want to implement include runway detection, airport detection, aerial object detection, object analysis and classification, landmark and ground feature detection, movement tracking, and trajectory prediction. All of these require the base object detection to be functional. From there post-processing the data from the neural network and cross checking additional data provided by the drone to get the results we want. Most of these features should be able to be implemented simultaneously due to our large team. We'll add additional features as time permits.

September

- Embedded System with Neural Network
- Nearly all training data acquired and prepared

October

- All General Functionality Finished (Database integration, all tools needed installed, etc)
- Object recognition working on board

November

- Testing in real world conditions
- Rough demo

7 Budget

Hardware:

- NVIDIA Jetson TK1 Development Kit - $299.99 (Based on research findings, will show if upgrade is required)
- oCam-1MGN-U Global Shutter Greyscale Camera - $126.00
- Micro-USB 3.0 Cable - $3
- 32GB SD Card - $12.33

Personel:

- Team V.O.L.D.E.M.O.R.T.

Total:

Base cost = $441.3

8 Conclusions

By project's close, we shall have a system utilizing an onboard system in realtime to provide information to flight systems regarding location of objects in a drone camera's view. We have some ambitious goals for this project, but we are dedicated and eager. We will have it detecting airports, providing useful information about objects it sees, and be able to be trained to detect new objects.

9 References

[1] A Machine Learning Approach to Visual Perception of Forest Trails for Mobile Robots
http://rpg.ifi.uzh.ch/docs/RAL16_Giusti.pdf

[2] Continuous online video classification with TensorFlow, Inception and a Raspberry Pi
https://medium.com/@harvitronix/continuous-online-video-classification-with-tensorflow-inception-and-a-raspberry-pi-785c8b1e13e1

[3] Five video classification methods implemented in Keras and TensorFlow
https://hackernoon.com/five-video-classification-methods-implemented-in-keras-and-tensorflow-99cad29cc0b5

[4] Board Information:
http://www.nvidia.com/object/jetson-tx1-module.html
http://www.nvidia.com/object/jetson-tk1-embedded-dev-kit.html
http://elinux.org/Jetson_TK1#Linux_distributions_running_on_Tegra

[5] Info on NIVIDIA:
https://www.nvidia.com/en-us/deep-learning-ai/developer/?ncid=pa-pai-cs27dl-4165&gclid=Cj0KEQiAnb3DBRCX2ZnSnMyO9dIBEiQAOcXYHyjKWcJ7ZjCZFupwJKSOS9VnycP0gkQi98WXt2RE7PsaAshs8P8HAQ

10 Appendices