Use of Imaging Devices and Machine Learning Software to Assist in Autonomous Vehicle Path Planning

DESIGN DOCUMENT

Team 3

SmartAg: Client

Joseph Zambreno: Advisor

Souparni Agnihorti: Meeting Facilitator

Ashley Dvorsky: Document Manager

Eric Himmelblau: Webmaster

Fahmida Joyti: Test Master

John Orefice: Communications Lead

Bowen Zhang: Hardware Maintainer

Team Website: sdmay18-03@iastate.edu

Version 2: Revised December 3rd, 2017

Table of Contents

List of Definitions

1. Single Shot MultiBox Detector (SSD): A unified framework for object detection with a single network.
2. DarkNet: A state-of-the-art, real-time object detection system.
3. DarkFlow: A superior version of DarkNet that is faster and more accurate.
4. Caffe: A deep learning open-source framework that is used with our SSD model to aid with image classification.
5. *mAP - Mean Average Precision - Mean of AP's in multiclass prediction
6. AP - Average precision - provides a measure of quality across all recall levels for single class classification. Seen as the area under the precision recall curve.

1. Introduction

1.1 Problem and Project Statement

SmartAg, an Iowa based startup, has developed an autonomous tractor system. This tractor is able to use a map of GPS coordinates to autonomously navigate a farm. However, this system works solely off of their precreated map, therefore the autonomous system is not able to react to real time changes in its environmental. If a farmer adds a fence after the map is made, the tractor will simply run it over. To solve this problem, SmartAg is proposing the use of an image recognition system which will work in tandem with the path planning map.

To accomplish this task, we plan to use stereo cameras to capture information about the tractor's environment in real time. The video from these cameras will be fed into an object detection system which will identify if there are any obstacles in the immediate environment. If an obstacle is detected, the system will determine its GPS coordinates from its distance relative to the camera, and add the obstacle to the map so it can be avoided in the future.

1.2 Operational Environment

This product will be working on mid-sized farms. It is exposed to all conditions including rain, dirt, and snow, but in moderation. If the weather is to the point that the conditions are not safe for the tractor or that the human observer would not be able to watch and effectively judge if they should deploy the emergency stop button this product will not be used. The exact details of the environment may vary, but the goal of this system is to allow the tractor to adjust accordingly.

1.3 Intended Users and uses

This product will be used by SmartAg in their autonomous tractor production with the end client's being large scale production farmers. Farmers, specifically farmers who have other time commitments which require their attention will found our product really useful. It greatly reduces the time spent by farmers on the farms for harvesting as the autonomous tractors can do it for them. As the tractor is being driving by the path planning software, it will detect obstacles through the stereo cameras, and then identify if there is a border or fence. Upon detecting and identifying an obstacle as such we will first start to navigate away from said obstacle

and then pass the GPS coordinates so the algorithm will avoid that obstacle in the future. It will also reduce costs and increase outputs for the farmers.

1.4 Assumptions and Limitations

1.4.1 Assumptions:

- The open source Convolutional Neural Network is working as intended
- The path planning algorithm provided by the client is working as intended
- The data acquired by the stereo camera system is accurate.
- Our tractor can see in at least 10 meters in front of them., otherwise an emergency stop will be deployed.

1.4.2. Limitations:

- The additional cost for cameras and other sensors is limited to $1000
- It will need to be powered by the tractor electrical system.
- Lack of a pre-defined dataset to train our model requiring image acquisition to get the data before we can start analysing the data.
- Testing limitation: Training of our model would be done during harvest season but we will be unable to test the model we create on the same conditions because the testing will be done during winter.

1.5 Expected End Product and Deliverables

At the end of the project, we should have a fully functional object detection system that can accurately and consistently identify all objects within the scope of our dataset. This includes variations on each object, so it should not be limited to detecting a specific type of fence. In addition, we should also have a depth measuring system that can reliably calculate the distance of an object relative to the tractor. It will then translate this location to a GPS coordinate which will be passed to the path planning algorithm put in place by SmartAg.

2. Specifications and Analysis

2.1 Design Specifications

2.1.1. Functional Requirements

- The image processing system shall be able to detect objects such as fences, ditches, and terraces in real time using a trained neural network.
- Use depth determination techniques to find how far away an object is so that the tractor can successfully circumvent them.
- Add object positions to the path planning map so that they can be avoided in the future.

2.1.2. Non-Functional Requirements

- The speed of the real-time object detection system must be at least 15 FPS.
- The system must fit within the cab of a late model tractor and not prevent manual driving.
- The system must be powered by the tractor electrical system.
- Needs to be usable by the target audience (farmers).

2.2 Proposed Design

We are using OpenCV[1], Python, and the MobileNet-SSD Neural Network[2] to aid with the image processing. We decided to use MobileNet-SSD after comparing it with two other neural networks - Darkflow[3] and Darknet[4]. SSD provides a more accurate object detection as well as faster performance when compared to Darkflow and Darknet making it a clear choice for our project. Additionally, SSD's primary disadvantages - difficulty detecting similar objects, and a lower accuracy when identifying small objects compared to large objects - will not have a major impact on our project as we will be detecting fences, ditches, and terraces which are large and dissimilar. We are manually acquiring images of fences, terraces and ditches from the web and using ImageNetUtil [5] to annotate these images and classify them into their respective categories. We are investigating ways of acquiring images and labels of interest from the ImageNet[6] and Open Images[7] datasets. Another area of focus we

[1] Reference to OpenCV
[2] Reference to MobileNet-SSD
[3] Reference to Darkflow
[4] Reference to Darknet
[5] Reference to ImageNet Util
[6] ImageNet
[7] Open Images

are investigating is the distance measurement system that we will use to determine how far away an object is. This will consist of a system of stereo cameras that will allow us to compare the relative location of an object in each of the stereo images and calculate its distance relative to the cameras. This is accomplished by using triangulation to calculate a depth map using the disparity of features between the two camera images. In addition, we will be using a GPS to find the coordinates of the object detected and add that to the path planning map provided by our client for future record.

2.2 Design Analysis

Our design analysis began by researching the different kinds of neural networks for object detection that were available and would suit our purposes. There were three primary neural networks we considered - Darkflow, Darknet and SSD. The decision to use SSD to do our object detection came after doing thorough research and analysis of all the neural networks. The main disadvantage of Darkflow and Darknet was that they had trouble identifying objects that had different aspect ratios and configurations, which could be a hindrance since it will need to recognize many variations of the same class of object. We would be training our neural network with pictures of fences, ditches and terraces whose size differs according to the type of farm. Fortunately, SSD did not have this disadvantage. In addition to that, since our dataset consisted of relatively large and dissimilar objects, we could override the main disadvantages of SSD which was that it had difficulty distinguishing between similar objects (ex horse vs zebra) and that it gave a lower accuracy when identifying small objects compared to large objects. A complete analysis of the advantages and disadvantages of the three neural networks has been made and can be referred to in Table 1 in the Appendix.

The training of our SSD neural network was done on the VOC0712 dataset which is a combination of the Pascal VOC2012[8] and the Pascal VOC2007 [9]datasets, which contains the object classes - Person, Animal (bird, cat, cow, dog, horse, sheep), Vehicle(aeroplane, bicycle, boat, bus, car, motorbike, train) and Indoor (bottle, chair, dining table, potted plant, sofa, tv/monitor). We tested it by inputting our own images and it was able to classify the images correctly with an accuracy between 97 - 100%. We used the Logitech - C920 Pro Webcam and ran the SSD in real-time. We were able to achieve a speed of 30FPS(the maximum achievable through that webcam) when we reduced the resolution of the output feed.

[8] VOC2007- http://host.robots.ox.ac.uk/pascal/VOC/voc2007/
[9] VOC2012 - http://host.robots.ox.ac.uk/pascal/VOC/voc2012/

Currently, we are acquiring a dataset containing objects of interest (fences, terraces and ditches.) We are using the ImageNetUtil tool to aid us with labeling the images that will be used for training SSD. We are exploring large sources like the ImageNet dataset from Stanford and the Open Images dataset from Google to retrieve more images and labels pertaining to our project.

We have decided to use stereo cameras for gathering images to classify. This decision was made after our client had issues with using a 360 degree camera and tilt shifting. Once the image from the camera is fed into our neural network and classified, we will use the triangulation features of OpenCV to calculate the distance of the obstacle using the stereo cameras as our point of reference. A GPS which is accurate under 2 inches will then be used to estimate the coordinates of our obstacle which will then be recorded to future navigation of the vehicle.

3. Testing and Implementation

3.1 Interface Specifications

- **Stereo cameras and image recognition system (hardware and software):**

  This interface is achieved using OpenCV (Open Source Computer Vision Library). OpenCV is an open source library with more than 2500 optimized algorithms to help us building our interfaces. We use OpenCV's functions to capture the video feed and separate the video to picture frames. These picture frames will have two functionalities. Firstly, the frames will be sent as inputs to the image recognition system to detect and classify the objects in front of the autonomous tractor. Secondly, the pictures captured by the two cameras at a particular time will be used for the distance determination calculations.

- **GPS and distance determination system (hardware and software).**

  This interface is built by SmartAg, and test results have shown that it is accurate enough to transfer data from GPS to the distance determination system to locate the autonomous tractors.

- **Image recognition system and distance determination system.**

  We will use python to transfer the objects found by image recognition system to distance determination system in order to match the objects and the its corresponding distance.

- **Path planner and distance determination system.**

  In the distance determination system, the distance and the angle of every object will be calculated with respect to the stereo cameras. Instantly, the coordinate of every object will be added to a path planning map made by our client which will in turn help us navigate the autonomous tractor.

3.2 Hardware and software

- **Hardware**
  - Nvidia TX2 GPU will be provided by our client, SmartAg, and is already tested by running different machine learning algorithms.
  - The StarFire 6000 GPS was already tested by SmartAg, and can send data to the Nvidia TX2.
  - Stereo cameras will be first tested indoor using fake objects. By compare objects' the actual distances and the calculated distances using the stereo cameras and object distance algorithms, we will know the accuracy. Second, we will tested the stereo cameras outdoors using real object. Last, we will install them on the autonomous tractors and test the accuracy.

- **Software**
    - MobileNet-SSD is an open source caffe implementation of Google's MobileNet SSD detection network. Compare to other frameworks, like YOLO, R-CNN, Fast R-CNN, it has a higher FPS to suit our hardware limitations.
    - ImageNet_Utils is used to, facilitate manually labeling images and creating label files in accordance with the requirements to train neural networks.

We will be writing automated tests using the Python UnitTest module. It is a common testing framework provided by Python and has similar behavior like Junit in Java. This will allow us to determine correct behavior for parts of our system. Also we will be using unittest.mock which will allow us to replace parts of our system with mock objects. Hence, we will be able to test our distance determination system without relying on the image recognition system as we can use mock objects instead. However, these tests will only allow us to catch defects in early stage of our development stage. We will do detailed tests to determine the accuracy of our system which is detailed more in the process section.

3.3 Testing Requirements

Our system will be separated into two primary parts: an object detection system and a distance calculation system. To ensure that our system is thoroughly tested, each subsystem will undergo a suite of individual tests, followed with integration testing to verify that they work properly together. Both individual and integration test suites will include a combination of functional and nonfunctional tests to confirm that all aspects meet the requirements set by SmartAg.

3.3.1 Object Detection and Classification

The two primary facets of the object detection system are its speed and accuracy when detecting objects of interest. These parameters are highly dependent on both the hardware and neural network used for this task, however the hardware is set by the client, so our testing will focus on the neural net. Initial testing will involve a preliminary comparison of openly available neural nets. Their speed, accuracy, and ease of use, including training and implementation, will be considered. Once a predominant neural network is chosen, it will be more thoroughly tested to ensure that its speed and accuracy of detection are sufficient for the autonomous tractor system, as detailed in *Section 2.1*.

3.3.2 Distance Measurement System

The evaluation criteria for the distance measurement subsystem is fairly straightforward. It must consistently measure the distance to an object, accurate

within a meter, solely using information from image data. However, there are many different methods of doing of calculating image distance. Initial testing will primarily be focused on deciding whether to use a single image or stereo images to achieve this measurement. Once a camera system is decided, further testing will be done to make certain that the distance can be found quickly and accurately enough to be used in a real time system.

3.3.3 Integration Testing

Once both the object detection and distance measurement subsystems have been individually tested, integration testing will be performed so confirm that the two can properly work together. There are a number of requirements for this testing to be successful. The most basic test will simply be to show that images from the camera(s) can be fed into the neural net for classification. Once this is confirmed, we will test the distance measurement systems ability to accurately calculate relative distance based on the information returned by the neural net. Finally, we will test the compatibility of this data with the path planning algorithm already in place by SmartAg.

3.4 Process

3.4.1 Object Detection and Classification

We will run our convolutional neural network on a training and test set that we make. Our system will be considered satisfactory if the neural net achieves a mAP score of at least 0.70 on the training set. That is, for every 10 images received, at least 7 would be classified correctly. Additionally, in order to minimize expected risk, we will try to minimise the loss function and attempt to keep it below 0.6. We are also aiming to achieve a minimum accuracy of 80% on the test set and have it run at a minimum rate of 15FPS in real-time when tested with the stereo cameras.

Additionally, while training and early testing can be performed on a desktop grade GPU, it is imperative that final testing is done on the Nvidia TX2 GPU so that the data can be extrapolated to real world situations. Once, we have achieved a sufficient level of accuracy on our training and test set we will move on to testing our model in conjunction with the distance determination system in real-time situations.

3.4.2 Distance Measurement

We will be using the same testing process as above to test our distance determination system. However, when testing the distance determination system our primary focus will be on how accurately it is able to determine distance to from

the camera point of view to the obstacle detected by the convolutional. We will start by putting a distance determination system in our hand and an obstacle front of us in 10 meters. Then we will move around the obstacle recording the coordinates provided by the distance determination system.Then we will compare the results given by the distance determination system with what we got in advance by measuring on own. We will repeat this experiment with obstacle at various distances to find the optimal range in which it is accurate. Additionally, at this stage of testing we will be able to determine how well our distance determination system integrates with our image recognition system. Next, we will proceed on to integrate our image recognition system with the tractor and the stereo cameras.

3.4.3 Integration Testing

For our system to be deployable in a real world environment, we need to be able to deliver a live video feed to the neural network and use its output as the input to the distance calculation system. This must then seamlessly be provided to the path planning algorithm. The integration of these various parts is crucial to the viability of our system.

The connection between the video feed and the neural net will be taken care of early on, using OpenCV. This connection will easily be verified if the neural net is able to classify objects from a live video feed. A more complex integration test will be required to confirm that the object detection system and distance measurement system properly work in tandem. To test this, we will first need to ensure that we can use the output of the neural net as an input for the distance measurement system. This means that both the image itself, and the information about the objects detected must be readable by the distance measurement system. If the distance measurement system, which should already have passed initial lab tests, can accurately calculate the distance to objects identified by the object detection system using this data, then these two subsystems will be successfully integrated. This will initially be tested in a controlled lab environment using well defined objects are preset distances. Once this test is passed, we will need to confirm that this relative location can be accurately combined with the GPS coordinate of the cameras to find the GPS location of the object. If the results are successful, the next phase of integration testing will begin.

With lab tests completed, field tests can begin. This will include setting the whole system in the tractor detecting images on a farm. The same steps mentioned above will be repeated, initially will the tractor sitting stationary, and eventually with the tractor actually running. Before the moving tests, we will need to verify that the GPS coordinates are successfully being added to the path planning algorithm. This

system will be tested against various objects of interest in multiple weather and lighting conditions to ensure that all parts are working properly.

3.6 Modeling and Simulation

Initial tests for the object detection and distance calculation systems will be performed in a lab environment. The object detection system is able to primarily be tested virtually while the image detection system will need physical models. After finishing all of our lab tests to determine that the image recognition and distance measurement systems are working accurately, we will proceed to simulate real world situations. We will do all of this as follows:

- Object Detection System (Lab Test)
    - Verify that Caffe, SSD and MobileNet-SSD are set up correctly by training it on the Pascal VOC2012 and the Pascal VOC2007 dataset
    - Test that it works with both pre-recorded and live video from a webcam
    - Train and test the MobileNet-SSD with the dataset that we acquire
        - Satisfactory results would include a mAP score of at least 0.70 on the training set and loss function of less than 0.6 on the training set.
        - Achieve a minimum accuracy of 80% on the test set and have it run at a minimum rate of 15FPS in real-time when tested with a webcam
- Distance Calculation System (Lab Test)
    - Provide the distance calculation system with stereo images of an object with bounding boxes
        - Initially, these images and bounding boxes will be selected manually
    - Compare calculated distance to manually measured distance
        - Error should be no greater than two feet
    - Repeat for objects of various sizes in different environments
    - Once results are satisfactory, repeat the steps above with images provided directly from the object detection system
- Field Testing
    - Manually drive the tractor on a farm with various obstacles for it to detect
    - Verify that the system consistently and accurately detects objects, calculates distance, and communicates the data to the path planning system

- If performance is satisfactory, test the tractor with autonomous driving
  - We will have a remote kill switch in case anything goes wrong
- Repeat above steps in different weather conditions and circumstances.

These tests will be done both with and without radar to assist in distance calculation to determine if radar is a necessary addition to the system. Results from testing will give us confidence as to which conditions the system will perform best in, as well as which conditions should be avoided.

3.7 Implementations Issues and Challenges

In the preliminary stages of our project, many compatibility issues have arisen. Much of the software that we are using is designed for use on Linux, while our group primarily owns Windows computers. We have gotten around this by using Linux computers at Iowa State University, however this has lead to additional issues regarding file permissions as well as storage available for saving image data.

One of the main issues we had was setting up and running SSD. The SSD would only run on an Nvidia GPU, so it could not be implemented on most of our personal computers. In addition, it required a framework called Caffe which was incredibly difficult to set up on windows computers. We got around it by using the Titan X box available in the Durham labs. This enabled us to easily install caffe and eventually train and test SSD after getting past some permission issues. One remaining issue with this solution however, lies in the graphics card. The Titan X GPU that we are using for testing is vastly more powerful than the TX2 GPU that we will use in production. This makes it challenging to extrapolate the test data to predict real world performance, which means that we will have to recreate our tests using production hardware as well.

One of the primary implementation issues that this project will face is simply gathering a sufficiently large data set to train and test the neural network used for object detection. This will require us to learn how to work with pre-existing data sets and supplement those by generating our own. Manually gathering and labeling images is a time consuming process, and could end up cutting into time allotted for other aspects of this project. Additionally, if we do not feel that we are able to gather a large enough set of images, we will have to research methods of altering images so that they can be used over again. To supplement this, we will also make use of the publicly available datasets: ImageNet, Open Images, and Field[10]SAFE.

[10] FieldSafe

Following this, we have had difficulty acquiring images from the FieldSAFE dataset as the it was created using the ROS operating system. Since, we had never worked with ROS before we had to become familiar with ROS topics and bag files. After getting familiar with ROS, we were able to obtain images from a 1 GB example file we got from the FieldSAFE dataset. However, the dataset we are really interested is of 112 GB size and we are unable to download it due to network timeout issues and storage size issues. Similarly, we have run into permissions issues when trying to access Google's Open Images dataset, which should be resolved by making an application to Google.

Beyond simply doing lab tests of our system, we will have to perform field tests as well. This poses additional challenges for us since we can only perform field tests at times that work for both our group and our client, SmartAg, since we are reliant on their tractors and access to farms. Other factors such as weather and snowfall can impose additional limitations to field testing.

4. Closing Material

4.1. Conclusion

Our group will be working with SmartAg to use image recognition software to detect obstacles that are commonly found in farming fields. Once an obstacle is detected we will return the location information to the path planning algorithm to avoid said obstacle and then also store the gps coordinates to the map to avoid said obstacle in future path plans.

As a team we have researched extensively the different options for making this project a success. We compared the benefits of 360° cameras, lidar sensors, stereo cameras to collect environmental data. We have concluded that the costs of the lidar sensors would be too far outside the budget of this project to be feasible, and the 360° camera had too many issues with tilt shifting and unwrapping affecting the quality and usability of that image. Because the low cost and few issues with image quality we have made the decision to use a stereo camera, which our client fully supports.

We will use OpenCV as our image recognition software. That is industry standard, and we have found for there to be a great number of reliable resources available to us. This is also what our client has been using so we believe this will help with integration of our final project as well as having the common language with our client who can be used as a resource.

After thorough research into our options we have made the decision to use mobile SSD for our neural network. That will give us the fastest computations as well as a seamless integration into OpenCV. There are many resources available and we have it the easiest to use.

4.2. References

1. OpenCV library. (n.d.). Retrieved November, 2017, from https://opencv.org/
2. Chuanqi. “MobileNet-SSD, GitHub, 5 Aug. 2017. https://github.com/chuanqi305/MobileNet-SSD
3. Thtrieu. “Darkflow.” GitHub, 24 Sept. 2017, https://github.com/thtrieu/darkflow.
4. Pjreddie. “Darknet”. GitHub. 1 Oct. 2017. https://github.com/pjreddie/darknet
5. Lin, T. T. (2017, March 10). Tzutalin/ImageNet_Utils. Retrieved from https://github.com/tzutalin/ImageNet_Utils

6. ImageNet. (n.d.). Retrieved from http://www.image-net.org/

7. O. (2017, November 28). Open images/dataset. Retrieved November, 2017, from https://github.com/openimages/dataset

8. The PASCAL Visual Object Classes Challenge 2007. (n.d.). Retrieved from http://host.robots.ox.ac.uk/pascal/VOC/voc2007/

9. The PASCAL Visual Object Classes Challenge 2012. (n.d.). Retrieved from http://host.robots.ox.ac.uk/pascal/VOC/voc2012/

10. FieldSAFE – Dataset for Obstacle Detection in Agriculture. (n.d.). Retrieved December 05, 2017, from https://vision.eng.au.dk/fieldsafe/

11. Zelener, A. (2017, July 02). Allanzelener/YAD2K. Retrieved November, 2017, from https://github.com/allanzelener/YAD2K

12. Balanca, P. (2017, April 10). SSD-Tensorflow. Retrieved November, 2017, from https://github.com/balancap/SSD-Tensorflow

13. "Feed the World Smarter." Smart Ag, https://www.smart-ag.com/.

4.3. Appendices

Table 1: Comparison of Available Neural Nets

Neural Network	Advantages	Disadvantages
DarkNet	- Runs at 45 FPS using an Nvidia Titan X GPU - Understands generalized representation of objects.	- Difficulty to predict objects in groups. - Loss function treats the error the same for small and large bounding boxes. -Has trouble when it sees unfamiliar aspect ratios or configurations
DarkFlow	-Runs at 67 FPS using an Nvidia Titan X GPU -Darknet predicted only 98 boxes per image but Darkflow predicts more than a thousand boxes per image - mAP of 78.6 on a 544 x 544 image - Recall of the convolutional neural network increases from 81% to 88% compared to DarkNet	- Difficulty determining two similar, but differently sized objects. - Still has difficulty predicting objects in group. - Still has trouble when it sees unfamiliar aspect ratios or configurations
SSD	- Significant improvement in speed for	- Relies on Caffe which is difficult to

	high accuracy detection compared to the others. - Runs at 59 FPS with 74.3% mAP when tested on the VOC2007 using an Nvidia Titan X GPU - Naturally handles objects of various sizes (due to the combination of different feature maps with different resolutions) -Can detect objects in multiple aspect ratios	set up on a Windows computer. - Difficulty distinguishing between similar objects (ex horse vs zebra) - Lower accuracy when identifying small objects compared to large objects