Object Detection and Identification with Sensor Fusion

PROJECT PLAN

#18
Client: Michael Olson - Danfoss
Advisor: Dr. Wang

Tucker Creger - Project Manager
Kellen O'Connor - Deep Learning Architect
Eric Bishop - Software Developer
Mitch Hagar - Radar System Lead
Clayton White - Hardware Design Engineer
Nihaal Sitaraman - Hardware Developer

sddec18@iastate.edu
http://sddec18-18.sd.ece.iastate.edu/

Revised: 04-22-18/3.0

Table of Contents

List of Figures

Figure 1: High-level Design Diagram

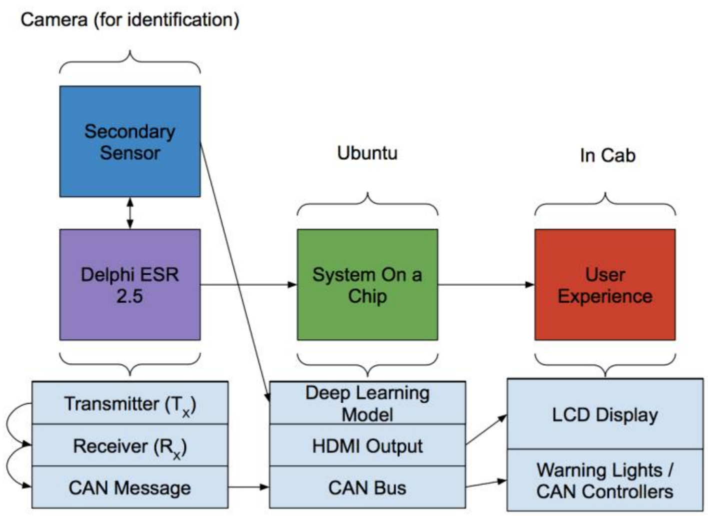


Figure 2: Design Process

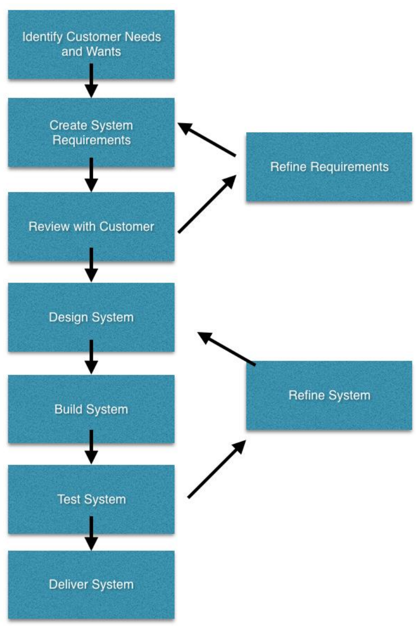
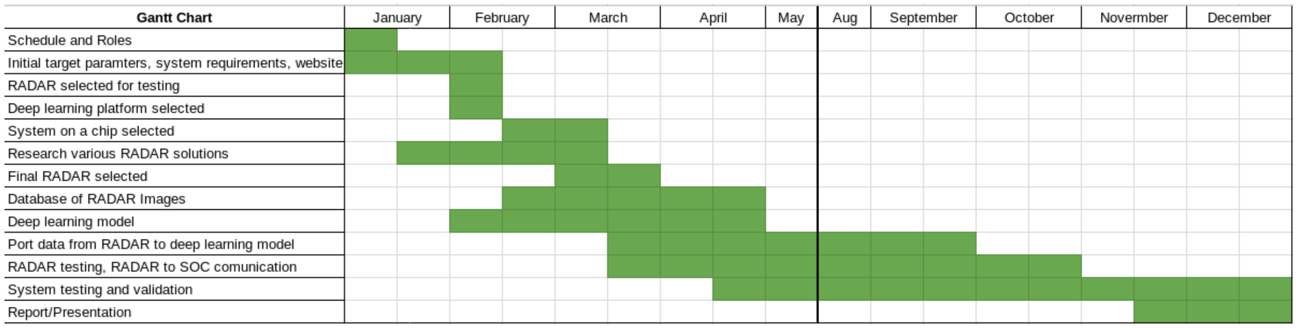


Figure 3: Gantt Chart of proposed schedule for the spring and fall semester.



List of Definitions

Radar: Radio Detection and Ranging

ESR: Electronically Scanning Radar

LIDAR: Light Detection and Ranging

SOC: System-on-a-Chip

LCD: Liquid Crystal Display

PEP: Python Enhancement Proposal

IEEE: Institute of Electrical and Electronics Engineers

RGB: Red, Green, Blue

SSD: Single Shot Detector

FRCNN: Faster Region-based Convolutional Neural Network

CNN: Convolutional Neural Network

CAN: Controller Area Network

HDMI: High Definition Multimedia Interface

API: Application Program Interface

1 Introductory Material

1.1 Acknowledgement

The project team would like to thank Michael Olson and Radoslaw Kornicki from Danfoss for their support on this project. We would also like to thank Dr. Wang for advising our team.

1.2 Problem Statement

As the agriculture and construction industries require autonomous solutions for increased safety and productivity, the need to sense objects in the equipment path increases. Examples include detecting people, animals, vehicles, and other large objects. There are many solutions on the market today using cameras and LIDAR. These solutions have limitations in weather and low light conditions. We were tasked with creating a system using radar to eliminate these limitations. Radar is able to operate in the dark, rain, snow, and fog. In some situations, radar can "see" through objects.

Our project consists of two main components: the implementation of a radar system and the development of a deep learning model. The radar system will allow us to detect objects in many different condition. The deep learning model will use a camera to identify objects in the equipment's path. This will allow for a notification to the equipment operator of objects in the vehicle's path and in the future, fully autonomous operation.

1.3 Operating Environment

The operating environment for the system will be on agriculture and construction equipment. This will require the system to be able to withstand water and dust from the operating environment and the vibrations associated with operation.

1.4 Intended Users and Intended Uses

The intended use is for certain agricultural vehicles and construction equipment that are a key area for our client. The long term use case is fully autonomous operation of machinery. It can also be used as an operator aid to increase safety.

1.5 Assumptions and Limitations

Assumptions:

- The operating conditions for the equipment will be normal and not abusive.
- The system will be mounted in an area that is protected from impact.
- The system-on-a-chip (SOC) and radar will be able to operate in a rugged environment.

Limitations:

- The system will only operate up to 15 mph. This will cover a large range of agriculture and construction equipment.
- The system will not be 100% immune to sensor blockage by dust and dirt.

1.6 Expected End Product and Other Deliverables

For this project, our deliverables are a whole system including a radar module and camera working with a deep learning model running on a SOC to perform object localization and classification. The system will notify the vehicle operator via an LCD screen in the cab of the objects' positions and types. The delivery date is December 2018. This system will be used for a demo on a piece of construction or agriculture equipment for our client.

Other deliverables include proposals regarding our radar and SOC selection, reports on which deep learning platforms are most suited for use in mobile applications, and a final report regarding the feasibility of implementing radar in construction and agricultural applications. The delivery date of all reports is December 2018. The delivery date of proposals to purchase the SOC and radar will be February 2018 and March 2018, respectively.

An itemized list of our deliverables is included below:

1. NVIDIA Jetson TX2 with attached LCD screen (LCD screen not property of Danfoss), camera, and Delphi ESR 2.5.
2. Trained deep learning model for object detection relevant to an agricultural/construction vehicle.
3. Python script that utilizes attached components and deep learning model to display object information on in-vehicle LCD screen.

Functional specifications are given in section 2.2.

2 Proposed Approach and Statement of Work

2.1 Objective of the Task

Our objective is to evaluate various radar technologies for Danfoss and through a combination of digital signal processing and deep learning, perform object detection and localization. By December 2018, we will have selected a radar option, a computing system adequate for a rough environment, designed and trained a neural network with data collected from the radar system and camera, and implemented it on a vehicle to alert an operator of the presence and location of unique objects via an LCD screen.

In order to provide value to Danfoss, we will also include a report evaluating various radar technologies, deep learning platforms, and computing systems to assist them in making a business decision when deciding to implement this technology in the future.

2.2 Functional Requirements

The functional requirements for the proposed design focus on robust detection and operation in agricultural and construction environments. A list of functional requirements is shown below.

1. The system shall have a range of 60 meters.
    a. Rationale: A range of 60 meters is required for early detection and identification. A machine traveling at 15 mph will cover 60 meters in under 10 seconds. This range allows for an object to be detected with sufficient time for action.

2. The system shall function on machines travelling at up to a speed of 15 mph or 6.7 m/s.
    a. Rationale: Most agriculture and construction equipment travel at speeds in the range of 5 mph to 15 mph. A max speed of 15 mph will allow for the majority of applications to be covered.

3. The system shall have angular range of ±30°.
    a. Rationale: An angular range of ±30° is required to detect objects in the vehicle's path with sufficient time to stop.

4. The system shall have a processing speed of 15 frames per second.
    a. Rationale: The system needs to detect an object with sufficient time to react. A frame rate of 15 frames per second on a vehicle traveling approximately 15 mph or 6.7 m/s means the vehicle will travel no further than 0.5 m between each frame update.
5. The system shall detect objects greater than 0.4 m size.

    a. Rationale: A width of 0.4 m is the width of human shoulders. Detection of a human is, at minimum, required for safe operation of the system.

6. The system shall be weather resistant to water, dust, and shock.
    a. Rationale: Danfoss' target applications involve heavy machinery that works in tough environments.

7. The system shall have a probability of missed detection less than 0.3.
    a. Rationale: A probability of 0.3 means that for each subsequent frame, the probability of missing an object multiple times approaches zero, which will yield a sufficiently short stopping distance.

8. The system shall have a probability of false alarm less than 0.3.
    a. Rationale: A false alarm, though undesirable, will be a safer alternative than a missed detection.

9. The system shall run off of a 12V power supply.
    a. Rationale: This voltage is easily available from a selection of batteries with also a range of amp hours. It is also easily available on a heavy equipment chassis. A step-up converter or inverter is acceptable.

10. The system shall fit inside 1'x1'x1' space.
    a. Rationale: Space is limited on a vehicle, so our design must be compact enough to not obstruct operator view or regular vehicle operation.

11. The system shall detect at least 4 classes of objects.
    a. Rationale: Our system should detect people, cars, construction/agriculture equipment, and buildings.

12. The system shall/should operate in the temperature range from -40 to 125 degrees Celsius.
    a. Rationale: This is a common operating range for automotive sensors.

2.3 Constraints Considerations

As part of the project, we must evaluate various radar options, deep learning platforms, object detection networks, and computing systems. Evaluation of these systems will be centered around the functional requirements, but the behavior of the full system cannot be known without implementing all combinations of each option. Therefore, a written report regarding behavior of individual components is necessary to justify our choices.

Because this project will ultimately lead to a business decision from Danfoss, cost of the system must be considered. We will strive to minimize cost, but not at the expense of our functional requirements.

Our code must be well commented and accessible to the client. The team will use Gitlab for version management of our software. This includes our neural network, data acquisition tools, and any low-level radar code.

Training data acquired during our project must be accessible to our client, yet secure. We will collaborate with Danfoss to ensure data collected (which may include imagery from their facilities) is secure to their internal standards.

For our code, group members will follow an agreed upon coding convention. Because we expect much of our code to be Python, we will follow PEP 8 - Style Guide for Python Code. This is not a formal standard that is required, but rather an organizational structure to ensure readability.

The IEEE Code of Ethics (IEEE) will help guide our project, and ensure that our work does not violate the health and safety of our members, equipment operators, or Danfoss employees. This is similar to number 1 in the IEEE code of ethics. We will ensure that any research performed is well documented and cited where necessary. This follows number 3 in the IEEE Code of Ethics This code of ethics is applicable to our project because it may eventually be used to prevent injury, so ethical violations could indirectly cause harm eventually. Overall, our system does not violate any ethical considerations from IEEE, and we will take care to ensure this is the case for the duration of our project.

Beyond the aforementioned standards (IEEE Code of Ethics, PEP 8, and internal Danfoss standards), our project will not violate any ethical considerations. We believe that accurately reporting the system's capabilities is the most significant task we can do, which falls in line with the IEEE Code of Ethics.

2.4 Previous Work And Literature

Literature surrounding the use of deep learning with radar focuses on either close-range object classification, or improvement of synthetic aperture radar.

A study from the University of St. Andrews showcases how a short range radar can be used to differentiate between various objects (Stewart). This study is encouraging in that it shows how radar waves may be reflected in a unique way from different objects, but it does not show long-range applications, which is a significant shortcoming.

A research paper from Radar Conference [1] shows how deep learning can be used to improve the digital signal processing aspect of radar for synthetic aperture radar. This is beyond the scope of our project, as our project centers more around performing deep learning on RGB imagery. Also, our system must perform real-time detection, not reconstruct an image later.

Literature more relevant to our project is related to object detection on RGB imagery. Several methods have been published that detail balances between speed and accuracy of neural networks for object detection. Towards Data Science describes several object detection methods using deep learning for us to evaluate. Of interest are Single-Shot Detector (SSD)[2] and Faster Region-based Convolutional Neural Network (FRCNN)[3]. These network topologies take different approaches to detect objects. FRCNN proposes regions where an object might be, and evaluates each region to determine where an object lies in an image. SSD analyzes an image with fixed bounding boxes around what the CNN determines are relevant features to determine where an object is. For our purposes, SSD may be a better option to explore due to its improved speed compared to FRCNN, but at the expense of accuracy. Both network topologies deal with image resolutions beyond what our radar is likely capable of producing, so we may need to explore techniques for interpolation.

As a team, we must develop a network that can perform object detection in real time with sufficient speed and accuracy that does not rely on these existing networks. Because the data we collect is processed to produce an image, we have the advantage of the option to incorporate raw signal data into our network, which is not an option for state of the art object detection networks for imagery.

2.5 Proposed Design

In the end, our designed system must be able to detect and identify an object up to 60 meters away, as long as the object is within the 60 degree angle of view. The radar will be a Delphi ESR 2.5, connected to an NVIDIA Jetson TX2 through CAN.

A block diagram of our proposed design is shown in figure 1.

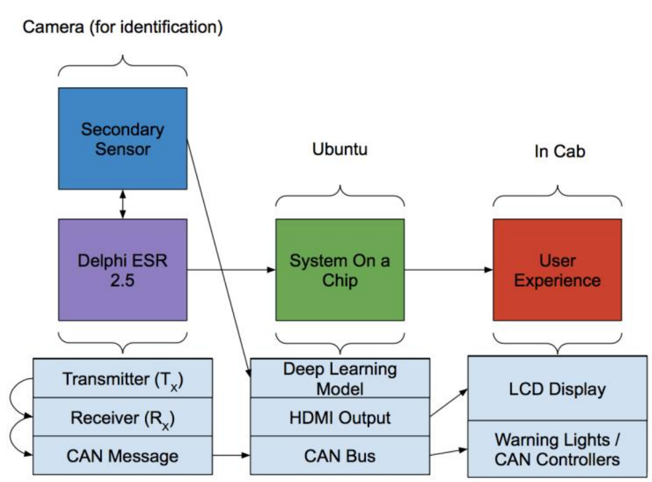


Figure 1: System Block Diagram

For the system output, the operator will be notified on an LCD screen within the cab, connected to the Jetson TX2 via HDMI.

The system must be able to identify the object. To do so, our deep learning model built with Keras will perform classification.

The system must be able to operate in all normal weather conditions such as cold, hot, windy, rainy, etc. The Delphi ESR 2.5, as an automotive radar, is well suited for the task.

Our proposed solution has several strengths and weaknesses, outline below:

Strengths:

- Redundant system for increased safety, as radar and camera will be used for detection
- Easy to understand output, because the operator will have a screen showing detected objects
- Robust against inclement weather because the radar can still see through fog when the camera cannot
- Improved functionality and cost compared to automotive LiDAR
- Deep learning model can determine object type with low computational overhead

Weaknesses:

- Limited field of view relative to LiDAR, which may present problems for objects moving quickly into the vehicle path from a shallow angle
- High cost relative to only vision. Cameras alone are significantly cheaper than radar
- Python is a slower language than C++, so implementation in the future would likely need to switch languages in order to improve performance
- An LCD takes up extra room in the vehicle cab

2.6 Technology Considerations

Many different radar systems and deep learning APIs are available. It has been decided that Keras is going to be the deep learning API for our system due to its simplicity, versatility, and a team member's previous experience with it.

A radar system by Walabot has been used as a test subject. This radar system does not have the range needed to meet design requirements, but it does come with an interface that is simple to operate. The system was used as a test platform to collect radar data as imagery.

Texas Instruments offers a chip and another supplier has a radar system that meets our distance requirements but does not offer an interface. Also, the system is quite expensive and exceeds our client's budget.

The Vayyar EVK is an attractive solution due to its relatively long range and ability to return an image. However, the increased cost and uncertainty about reliability in agricultural and construction environments led us to the Delphi ESR 2.5.

The Delphi ESR 2.5 was chosen as our final radar due to its simplicity to use. It returns the distance and angle to all objects in a series of CAN messages, which may be interpreted from the NVIDIA Jetson TX2. Although this radar does not provide an image we can use, we will utilize a camera to perform object classification for all detected objects.

2.7 Safety Considerations

If soldering circuits becomes a task, burns are a possible risk. Typical solder temperatures range from 500-800 degrees Fahrenheit. The team member attempting to use a soldering station should have basic knowledge on how it operates.

Our design will eventually be tested on large machinery such as agriculture and construction equipment which will create safety risks. Personal protective equipment (PPE) and/or training may be required in order to avoid cuts, head trauma, slips, etc.

When testing the system, the vehicle selected must have a secondary seat. The secondary seat will be used for a group member to analyze our system's performance. This will allow the vehicle operator's full attention to be on safely operating the vehicle.

2.8 Task Approach

Our task approach for the design can be visualized using the block diagram below. Early in the semester, we met with the client and determined system requirements. Next, the team will begin reviewing those requirements and designing the system. Once the radar system, SOC, and smaller sub-components have been obtained, the building process of the system will begin. Continued software revision and testing will constitute the bulk of Fall 2018's work.

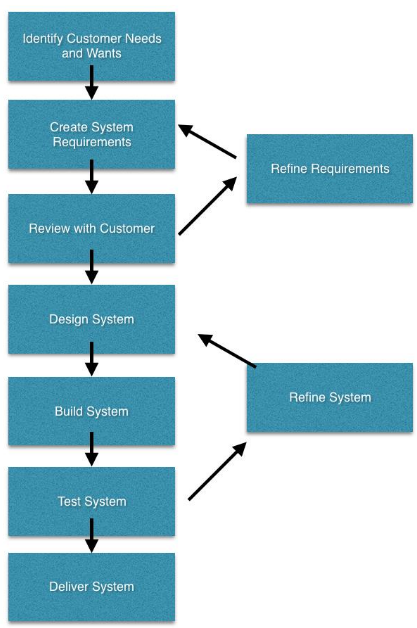


Figure 2: Design Process

2.9 Possible Risks And Risk Management

The cost of our system, in total, will reach at least $6475, The Delphi ESR 2.5 costs $6175 while the NVIDIA Jetson TX2 cost $300. Due to the high cost of the system, it has taken longer than expected to receive these necessary items. Other costs include a camera, connectors for various subsystems and CAN controllers.

Another risk is the time spent to train our neural network properly. Each neural network revision may require a trip to the Danfoss test track to test, or at the very least a significant time to retrain it. In order to deal with this risk, we will attempt to record a significant amount of test data (videos and radar logs) to verify our systems performance.

The performance of our deep learning model is crucial, so we need to ensure it is able to detect a variety of classes in multiple scenarios. To ensure it performs well in these scenarios, we need to gather training data from similar situations. Examples include cloudy weather, rain, potentially fog, and various times of day. Therefore, we must schedule data collection days early in the semester, yet be flexible in order to gather data from multiple weather conditions.

Aside from the aforementioned risks, technical challenges will be the biggest issue for our team to overcome. To manage technical challenges, we will collaborate with each other as well as reach out to Danfoss and Iowa State faculty members when we encounter issues. To prevent an issue from recurring, steps taken to solve an issue will be documented in our shared Google Drive for future reference.

2.10 Project Proposed Milestones and Evaluation Criteria

Deciding on a particular radar system is our first key milestone. This radar will need to meet all functional requirements. When we believe we have found the correct radar we will purchase it. The next milestone will be hooking the system up. Once we have the system setup we will test to make sure all components work with each other by testing the input and output of the system. As long as information gets from input to the output we will know we have it set up correctly. The last key milestone will be a working deep learning model. Our final product will be tested on our client’s test track in Ames. If our product detects the object it is supposed to, then we know it works. The test plan for this is in section 2.13.

2.11 Project Tracking Procedures

Our group is using GitLab and Google Drive to track our process. Tasks will be assigned to individuals for tracking in GitLab, and all relevant documents will be shared on Google Drive to ensure all group members have access.

2.12 Expected Results and Validation

Our desired outcome is to be able to detect four different classes of objects. Not only do we want to detect these objects, but we want the driver to be able to see where the object is and what the object is via a user interface.

We will validate each requirement at different levels of testing. This will ensure that our system is performing to specifications.

In order to meet each functional requirement, a test will be performed for each of the following requirements:

1. The system shall have a range of 60 meters.
    a. Criteria for success: We will consider the test a success if the system is able to detect objects with a false alarm probability of less than 0.3 and a missed detection probability of less than 0.3 at all tested ranges.
    b. Test: Range Test

2. The system shall function on machines travelling at up to a speed of 15 mph or 6.7 m/s.
    a. Criteria for success: We will consider the test a success if the system is able to detect objects with a false alarm probability of less than 0.3 and a missed detection probability of less than 0.3 at all tested speeds.
    b. Test: Speed Test

3. The system shall have angular range of ±30°.
    a. Criteria for success: We will consider the test a success if the system is able to detect objects with a false alarm probability of less than 0.3 and a missed detection probability of less than 0.3 at all tested angles.
    b. Test: Angle Test

4. The system shall have a processing speed of 15 frames per second.
    a. Criteria for success: We will consider the test a success if the system is able to have a processing speed of 15 frames per second.
    b. Test: Processing Speed Test

5. The system shall detect objects greater than 0.4 m size.
    a. Criteria for success: We will consider the test a success if the system is able to detect an object of 0.4 meters wide at 60 meters.
    b. Test: Range Test

6. The system shall be weather resistant to water, dust, and shock.

a. Criteria for success: We will consider the test a success if the components we selected have the appropriate IP rating and do not fail during full system testing on a machine.
b. Test: This requirement will be tested in several on-machine tests.

7. The system shall have a probability of missed detection less than 0.3.
    a. Criteria for success: We will consider the test a success if the probability of missed detection is less than 0.3
    b. Test: Missed Detection and False Alarm Test

8. The system shall have a probability of false alarm less than 0.3.
    a. Criteria for success: We will consider the test a success if the probability of false alarm is less than 0.3
    b. Test: Missed Detection and False Alarm Test

9. The system shall run off of a 12V power supply.
    a. Criteria for success: We will consider this requirement met if the system is able to run off a 12V supply with or without a boost converter or inverter.
    b. Test: This requirement will not be tested.

10. The system shall fit inside 1'x1'x1' space.
    a. Criteria for success: We will consider this requirement met if the radar module is able to fit in the required space and if the enclosure for the SOC is able to fit in the required space.
    b. Test: This requirement will not be tested.

11. The system shall detect at least 4 classes of objects.
    a. Criteria for success: We will consider this requirement met if the system is able to detect at least 3 classes of objects.
    b. Test: Object Classes Test

12. The system shall/should operate in the temperature range from -40 to 125 degrees Fahrenheit.
    a. Criteria for success: We will consider this requirement met if the components are rated for operation in the specified temperature range.
    b. Test: This requirement will not be tested.

The test plan in the next section describes in detail how we will administer the tests to validate the requirements.

2.13 Test Plan

We will conduct several different levels of testing. We will conduct testing at just a software level of the deep learning model. We will conduct component level bench testing and whole system bench testing. We will conduct whole system level testing with a stationary mount. Lastly, we will conduct whole system level testing on a machine.

1. Range Test
    a. We will validate the range requirement with testing of objects at 5, 10, 15, 20, 25, 35, 45, 55, and 65 meters. We will conduct this test with the system stationary and with the system moving on a machine at rated speed.

2. Speed Test
    a. We will validate the max operating speed by testing the system moving at 2, 5, 10, and 15 mph.

3. Angle Test
    a. We will validate the field of view requirement by testing the system with objects at different angles. We will test at 0°, 5°, 10°, 15°, 20°, 25°, and 30° with relation to the center line of the machine. We will conduct this test with objects at 5, 10, 15, 20, 25, 35, 45, 55, and 65 meters.

4. Processing Speed Test
    a. We will validate the frame rate by recording data for a set period of time and verifying that the number of frames corresponds to an average of 15 frames per second.
5. Missed Detection and False Alarm Test
    a. We will validate the probability of Missed Detection and False Alarm be verifying that the number of objects detected in a certain time period corresponds to a range of .7 times the real number of objects to 1.3 times the real number of objects.

3 Project Timeline, Estimated Resources, and Challenges

3.1 Project Timeline

A breakdown of our tasks for the first semester is given below for 8 two week sprints. We anticipate changes in our schedule as we progress, so a comprehensive sprint plan has not been created for second semester. We do, however, have a Gantt chart for both semesters.

Sprint #	Dates	Deliverables
1	01/08 - 01/21	Schedule and Roles: We will be solidifying our schedule, roles, responsibilities, and allotting times for meetings. We will be discussing individual positions and we will be assigning tasks and deadlines for specific portions of the project.
2	01/22 - 02/04	Target parameters, system requirements, and website: We will be working on improving our website. We will continue to updates documents and responsibilities, which will be published. We will compile a "system requirements" to know what hardware we require to run processes. We will also be solidifying our choice for Radar during this time so we may begin testing.
3	02/05 - 02/18	Final Selections: Our team will have decided on which radars are suitable, the deep learning platform, and what kind of system on a chip (SOC) we will be using as our onboard computer. Once selected, we can begin the testing process to narrow down a final combination of the 3 components.
4	02/19 - 03/04	Testing: In this portion of our project, we will actually begin to run tests on our final radar/platform/SOC combination to see how it fares and to see if the real world results are what we expected. If our expectations are met or exceeded, we will proceed and begin to prep out deep learning model and run identification tests.
5	03/05 - 03/25	Final Radar and SOC selected: This is the two week slot we have allotted in case we need to rethink our radar/platform/SOC combination. If we are satisfied with our primary combination then we will use these two weeks to begin training.
6	03/26 - 04/08	Database of Radar Images: As we approach the end of our semester, the team will start collecting data and classifying it to run it through our deep learning platform so we can identify various radar signals as objects.
7	04/09 - 04/22	Deep learning model: Once the database is complete we will run it through the deep learning model we selected. When it is ready, we can begin to load it onto our SOC to test it with real objects in front of it.
8	04/23 - 05/04	Port data from radar and camera to deep learning model: We will send the collection of data from the radar to the SOC to be run through the neural network. This way we can improve the accuracy of the net while building a bigger database.

A Gantt chart illustrating our plan for both semesters is shown below:

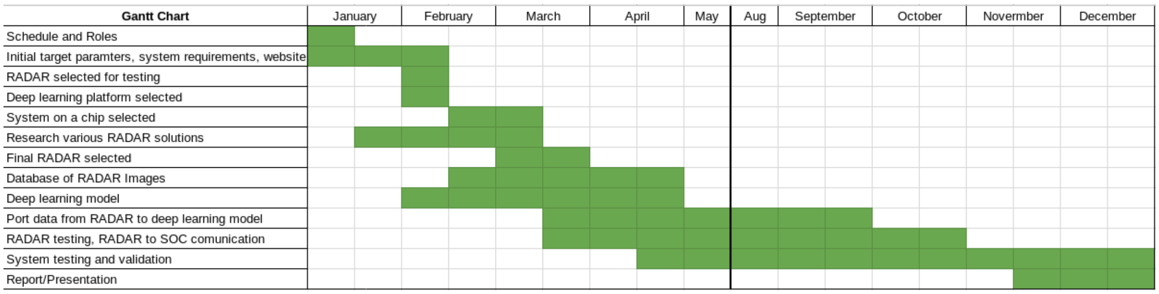


Figure 3: Gantt Chart of proposed schedule for the spring and fall semester.

3.2 Feasibility Assessment

Our project has several challenges that we will need to overcome. The first is managing cost. The Delphi ESR 2.5 is $6175, while our NVIDIA Jetson TX2 is $300. Smaller costs, such as CAN controllers, a camera, and connectors must be considered as well. The camera will likely be a <$100 webcam in order to demonstrate functionality. Connectors and associated tools will be from Deutsch, and will cost approximately $900. Demonstrating to Danfoss the value of the system as a prototype is the first hurdle that must be overcome - therefore we will create written reports justifying component choices and associated costs.

The feasibility of designing and training an object detection deep learning model is high. Object detection is a well-researched field in deep learning, and many resources exist to learn from. Combining the deep learning model output with the output of the radar will be the largest risk here. In order to keep this feasible, we will design our software in such a way that we warn the operator of a risk if seen by the radar and use the neural network to enhance information. An example is identifying the object, giving it a risk rating, and drawing a correct size bounding box around the object on the LCD. We can also use the neural network for non-maximum suppression to decrease duplicate hits for large objects detected by the Delphi ESR 2.5

Training the neural network and testing in a variety of conditions is one of the larger risks we will face. We must collect data from a variety of weather conditions and lighting scenarios in order to prevent our model from overfitting. To do so, we need to collect training data and testing data from as many different days as possible. In general, neural networks will perform better if given more data to learn from during training.

3.3 Personnel Effort Requirements

This information is found on our GitLab page[4]. Each action item includes dates, assignees, and details all listed out on the Issue Board section of our Git.

3.4 Other Resource Requirements

Our group must get a general understanding for radar, deep learning, and Python in order to effectively make this project work. To learn about each of these, we will utilize resources publically available on the internet and library. Examples include a variety of machine learning blogs and programming examples for Python found online.

To collect training data, we need access to heavy machinery and a testing area. Danfoss will provide this at their Ames facility.

Storage of training data must follow Danfoss' data privacy rules. If our client wishes for us to store data collected from their testing area on their servers, we will do so.

3.5 Financial Requirements

This system has several associated costs. Because this system is a single-quantity prototype, the cost of associated components will be high relative to volume production. A breakdown of costs is listed below.

- Delphi ESR 2.5: $6175
- NVIDIA Jetson TX2: $300
- Camera: <$100
- Deutsch tools, connectors: $900
- CAN Controllers: $13
- Enclosure for Jetson and circuitry: $30

4 Closure Materials

4.1 Conclusion

Our project is to develop a system for Danfoss to use on machinery such as tractors, loaders, excavators, and other heavy equipment that can use radar to detect objects. We will be utilizing deep learning to recognize objects in order for these vehicles to determine if there is a hazard in range of the radar.

We decided to use the Delphi ESR 2.5 because of its better range and wider field of view. This radar will be connected along with a camera to the Jetson TX2 computer which will then process the data and output information to the LCD monitor in the cab of the machine. We hope to create a device that we can put into testing on Danfoss' test track in Ames in order to show leadership and engineers from Danfoss the work we have accomplished and the system we have created.

With our collected data from this radar use deep learning to help decipher whether the objects in the view of the radar are a hazard or are not a hazard. This will then appear on the LCD with the classification of the object, and whether this object will be in the path of the vehicle or just a general potential problem.

With all of this accomplished we will have tested a new technology for Danfoss. This technology can help make safer working conditions. Hopefully, this technology will be feasible for future development into a product for our client, and a safer society surrounding the machinery our system is implemented on.

4.2 References

[1]E. Mason, B. Yonel and B. Yazici, "Deep learning for radar," *2017 IEEE Radar Conference (RadarConf)*, Seattle, WA, 2017, pp. 1703-1708. doi: 10.1109/RADAR.2017.7944481

[2]W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C. Fu, A. C. Berg. “SSD: Single Shot MultiBox Detector.” *Cornell University Library*, 29 Dec. 2016, https://arxiv.org/abs/1512.02325.

[3]S. Ren, K. He, R. Girshick, J. Sun. “Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks.” *Cornell University Library,* 6 Jan. 2016, https://arxiv.org/abs/1506.01497.

[4] Iowa State ECE Senior Team. “Senior Design GitLab Repository.” *GitLab,* 23 Apr. 2018, https://git.ece.iastate.edu/sd/sddec18-18

Stewart, Louis. “RadarCat for Object Recognition.” *SACHI*, sachi.cs.st-andrews.ac.uk/research/interaction/radarcat-exploits-googles-soli-radar-sensor-for-object-and-material-recognition/.

Van Rossum, Guido. “PEP 8 -- Style Guide for Python Code.” *Python.org*, 1 Aug. 2013, www.python.org/dev/peps/pep-0008/.

Xu, Joyce. “Deep Learning for Object Detection: A Comprehensive Review.” *Towards Data Science*, Towards Data Science, 11 Sept. 2017, towardsdatascience.com/deep-learning-for-object-detection-a-comprehensive-review-73930816d8d9.

4.3 Appendices

Delphi Electronically Scanning RADAR

Delphi has applied more than 20 years of radar experience to develop its award-winning electronically scanning radar (ESR). Leveraging expertise gained from radar production that began in 1999, Delphi brought ESR to market at a price that is helping make radar-based safety and convenience systems more affordable in the automotive market.

Delphi's multimode ESR combines a wide field of view at mid-range with long-range coverage to provide two measurement modes simultaneously. While earlier forward looking radar systems used multiple beam radars with mechanical scanning or several fixed, overlapping beams to attain the view required for systems like adaptive cruise control, Delphi's multimode ESR provides wide coverage at mid range and high-resolution long-range coverage using a single radar. Wide, mid-range coverage not only allows vehicles cutting in from adjacent lanes to be detected but also identifies vehicles and pedestrians across the width of the equipped vehicle. Long-range coverage provides accurate range and speed data with powerful object discrimination that can identify up to 64 targets in the vehicle's path.

Delphi's technologically advanced ESR uses proven solid state technology plus class-leading performance, packaging and durability to offer customers game-changing forward radar detection. The quality of data provided by Delphi's system enables powerful functionality including adaptive cruise control, forward collision warning, brake support and headway alert.

ESR enables the following features:

- Adaptive Cruise Control with Stop & Go
  - Enhances driver convenience
- Forward Collision Warning
  - Helps reduce the potential for an accident and injury
  - Helps reduce the potential for property damage
- Brake Support
  - Helps reduce the potential for an accident and injury
  - Helps reduce the potential for property damage
- Headway Alert
  - Provides distance information
  - Alerts driver when the preset time-gap to vehicle ahead is violated

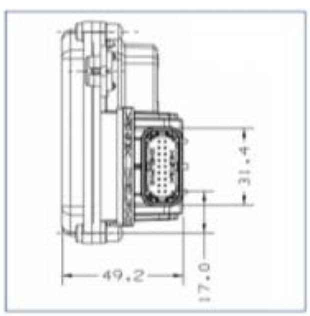
AUTONOMOUSTUFF
Tel. 309.291.0966 | www.AutonomouStuff.com | info@AutonomouStuff.com

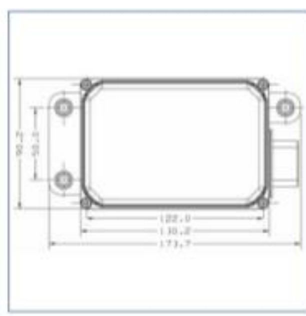
Delphi Electronically Scanning RADAR

The Delphi Advantage

- Multi-mode, multi-application capability
  - Simultaneous long- and mid-range functionality allows one radar to be used for multiple safety systems including adaptive cruise control, headway alert, collision warning and mitigation and brake support
- Solid-state Technology
  - No moving parts
  - Extremely reliable
- Class-leading performance and durability
  - Resistant to vibration and extremely robust
  - Innovative design provides excellent multi-target discrimination plus precise range, approach speed and angle data
  - Dual-mode classification enhances object reliability
  - Simultaneous Transmit and Receive Pulse Doppler (STAR PD) Waveform provides independent measurements of range and range-rate and superior detection of clustered stationary objects
- Compact packaging
  - Complete radar module, including electronics, measures just 173.7 x 90.2 x 49.2 millimeters including mounting features
  - Compact design makes it easier to locate the sensor on the vehicle without compromising vehicle styling
- High value
  - Produced using processes proven in high-volume manufacture of engine control units
  - Proven manufacturing processes increase affordability for high-volume automotive segments where radar systems have not previously been available

The radar module, including electronics, measures just 173.7 x 90.2 x 49.2 millimeters including mounting features.





Description		Jetson TX2 System-on-Module[*]
Pascal GPU [◊]		
256-core GPU | End-to-end lossless compression | Tile Caching | OpenGL® 4.5 | OpenGL® ES 3.2 | Vulkan® 1.0 | CUDA® 8.0 | GPGPU		
Maximum Operating Frequency		1.12GHz
CPU Complex [‡]		
ARMv8 (64-bit) heterogeneous multi-processing (HMP) CPU architecture; two CPU clusters (6 processor cores) connected by a high-performance coherent interconnect fabric. NVIDIA Denver 2 (Dual-Core) Processor: L1 Cache: 128KB L1 instruction cache (I-cache) per core; 64KB L1 data cache (D-cache) per core | L2 Unified Cache: 2MB ARM® Cortex® -A57 MPCore (Quad-Core) Processor: L1 Cache: 48KB L1 instruction cache (I-cache) per core; 32KB L1 data cache (D-cache) per core | L2 Unified Cache: 2MB		
Maximum Operating Frequency per Core NVIDIA Denver 2 ARM Cortex-A57		 2.0GHz 2.0GHz
HD Video & JPEG		
Video Decode(Number of Streams Supported): H.265 [(†)]: Main 10, Main 8 H.265 [(†)]: Main 444 H.264 [(†)]: Baseline, Main, High H.264 [(†)]: MVC Stereo (per view) VP9 [(†)]: Profile 0 (8-bit) and 2 (10 and 12-bit) VP8: All MPEG1/2: Main MPEG4: SP/AP VC1: SP/MP/AP		 (2x) 2160p60 | (4x) 2160p30 | (7x) 1080p60 | (14x) 1080p30 2160p60 | (2x) 2160p30 | (3x) 1080p60 | (7x) 1080p30 (2x) 2160p60 | (4x) 2160p30 | (7x) 1080p60 | (14x) 1080p30 2160p60 | 2160p30 | 1080p60 | 1080p30 (2x) 2160p60 | (4x) 2160p30 | (7x) 1080p60 | (14x) 1080p30 2160p60 | (2x) 2160p30 | (4x) 1080p60 | (8x) 1080p30 2160p60 | (2x) 2160p30 | (4x) 1080p60 | (8x) 1080p30 (4x) 1080p60 | (8x) 1080p30 (2x) 1080p60 | (4x) 1080p30
Video Encode (Number of Streams Supported): H.265 H.264: Baseline, Main, High WEBM VP9 WEBM VP8		 2160p60 | (3x) 2160p30 | (4x)1080p60 | (8x) 1080P30 2160p60 | (3x) 2160p30 | (7x) 1080p60 | (14x) 1080p30 2160p30 | (3x) 1080p60 | (7x) 1080p30 2160p30 | (3x) 1080p60 | (6x) 1080p30
JPEG (Decode & Encode)		600 MP/sec
Audio Subsystem		
Industry-standard High Definition Audio (HDA) controller provides a multi-channel audio path to the HDMI interface | 4 x I2S | DMIC | DSPK | 2 x I and Q baseband data channels | PDM in/out		
Display Controller Subsystem		
Support for DSI, HDMI, DP and eDP | Two multi-mode eDP/DP/HDMI outputs.		
Captive Panel		
MIPI-DSI (1.5Gbps/lane)	Max Resolution	Support for Single x4 or Dual x4 links | 2560x1600 at 60Hz
eDP 1.4 (HBR2 5.4Gbps)	Max Resolution	3840x2160 at 60Hz
External Display		
HDMI 2.0a/b (6Gbps)	Max Resolution	3840x2160 at 60Hz
DP 1.2a (HBR2 5.4 Gbps)	Max Resolution	3840x2160 at 60Hz
Imaging System		
Dedicated RAW to YUV processing engine process up to 1.4Gpix/s | MIPI CSI 2.0 up to 2.5Gbps (per lane) | Support for x4 and x2 configurations (up to 3 x4-lane or 6 x2-lane cameras)		
Clocks		
System clock: 38.4 MHz | Sleep clock: 32.768 KHz | Dynamic clock scaling and clock source selection		
Boot Sources		
Internal eMMC and USB (recovery mode)		

Description	Jetson TX2 System-on-Module*
Security	
Secure memory with video protection region for protection of intermediate results | Configurable secure DRAM regions for code and data protection | Hardware acceleration for AES 128/192/256 encryption and decryption to be used for secure boot and multimedia Digital Rights Management (DRM) | Hardware acceleration for AES CMAC, SHA-1, SHA-256, SHA-384, and SHA-512 algorithms | 2048-bit RSA HW for PKC boot| HW Random number generator (RNG) SP800-90 | TrustZone technology support for DRAM, peripherals | SE/TSEC with side channel counter-measures for AES | RSA-3096 and ECC-512/521 supported via PKA	
Memory ††	
128-bit DRAM interface | Secure External Memory Access Using TrustZone Technology | System MMU	
Memory Type	4ch x 32-bit LPDDR4
Maximum Memory Bus Frequency (up to)	1866MHz
Memory Capacity	8GB
Storage	
eMMC 5.1 Flash Storage	
Bus Width	8-bit
Maximum Bus Frequency	200MHz (HS400)
Storage Capacity	32GB
Connectivity	
WLAN	
Radio type	IEEE 802.11a/b/g/n/ac dual-band 2x2 MIMO
Maximum transfer rate	866.7Mbps
Bluetooth	
Version level	4.1
Maximum transfer rate	3MB/s
Networking	
10/100/1000 BASE-T Ethernet | IEEE 802.3u Media Access Controller (MAC) | Embedded memory	
Peripheral Interfaces ∆	
XHCI host controller with integrated PHY: (up to) 3 x USB 3.0, 3 x USB 2.0 | USB 3.0 device controller with integrated PHY | 5-lane PCIe: two x1 and one x4 controllers | SATA (1 port) | SD/MMC controller (supporting eMMC 5.1, SD 4.0, SDHOST 4.0 and SDIO 3.0) | 5 x UART | 3 x SPI | 8 x I²C | 2 x CAN | 4 x I2S: support I²S, RJM, LJM, PCM, TDM (multi-slot mode) | GPIOs	
Operating Requirements ♦	
Temperature Range	-25C – 80C
Module Power	7.5W (Max-Q) / 15W (Max-P)
Power Input	5.5V – 19.6V
Applications	
Intelligent Video Analytics, Drones, Robotics, Industrial automation, Gaming, and more.	

- * Refer to the software release feature list for current software support.
- ◊ GPU Maximum Operating Frequency: 1.3GHz supported in boost mode.
  Product is based on a published Khronos Specification and is expected to pass the Khronos Conformance Process. Current conformance status can be found at www.khronos.org/conformance.
- ‡ CPU Maximum Operating Frequency: 1-4 core = up to 2.0GHz; greater than 4 core = up to 1.4GHz
- (†) For max supported number of instances: bitrate not to exceed 15 Mbps per HD stream (i.e., 1080p30), overall effective bitrate is less than or equal to 240 Mbps
- †† Dependent on board layout. Refer to Interface Design Guide for layout guidelines.
- ∆ Refer to the Interface Design Guide and *Parker Series SoC Technical Reference Manual* to determine which peripheral interface options can be simultaneously exposed.
- ♦ Refer to the *Jetson TX2 OEM Product Design Guide* and *Jetson TX2 Thermal Design Guide* for evaluating product power and thermal solution requirements. See the software documentation for information on changing the default power mode (default: Max-P).