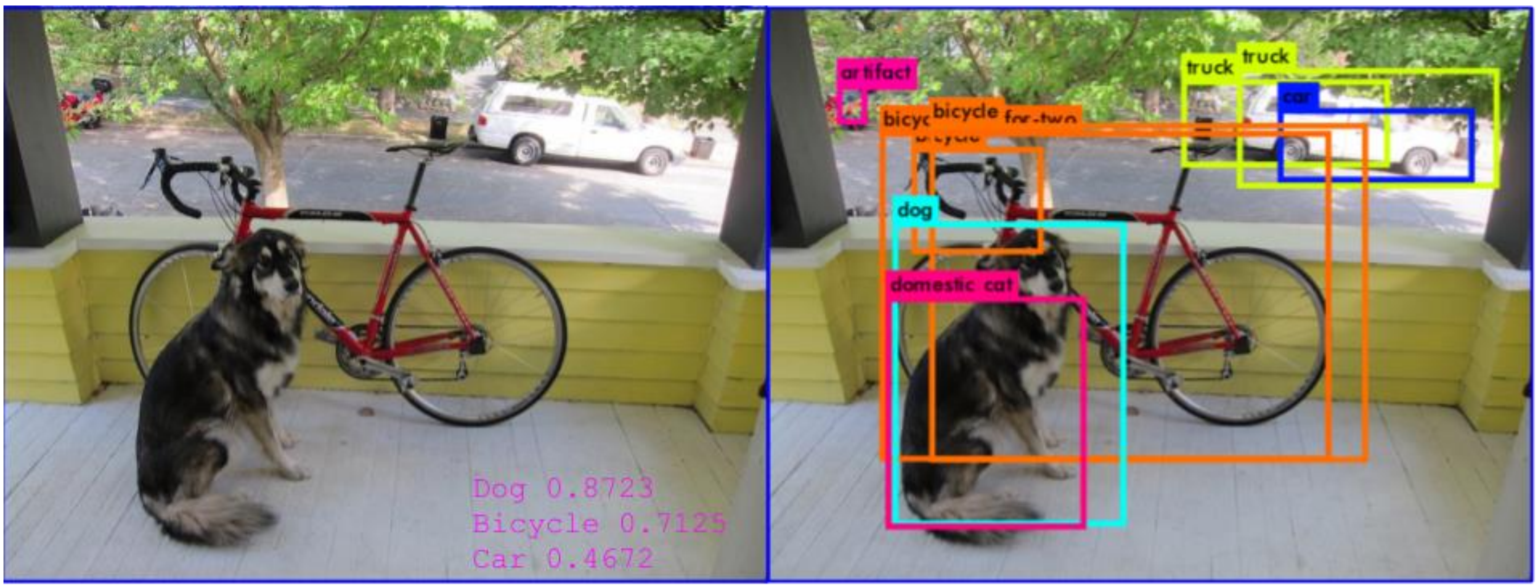
A STABLE REAL-TIME OBJECT DETECTION SYSTEM FOR UAV TECHNOLOGY



(Source: Jain, 2018.)

Robert Bara ENGR 2996:001, Technical Communication November 17, 2020

Abstract

This document discusses real-time object detection for drone/UAV technology to create a fast-stable system which can be easily implemented for various tasks. Three solutions are analyzed consisting of combinations of Visual and Radar/LiDAR detection. Throughout the solutions, two microcontrollers were compared the Jetson TX2 and RPi4 based upon their performances running various CNNs and their compatibility with their corresponding Radar/LiDAR components. Two solutions consisted of using LiDAR, while one solution used Radar, but based upon performance, cost, weight, and FCC approval, a LiDAR solution would be more beneficial to the overall project. From this, the Jetson TX2 solution in conjugation with RPLiDAR A2 solution was selected over the RPi4 and Velodyne LiDAR, for its CNN performance at 46 FPS with an accuracy of 84.2% due to the power of the TX2's GPU, integration with ROS accessibility, open source software provided by NVIDIA, and collaboration with the RPLiDAR A2. The proposed system primarily stems from research developed by Yuthika Sagarage, who further developed the CNN/LiDAR system for an autonomous drone, meaning the proposed solution additionally can be further developed into an autonomous drone with the help of the ROS interface.

Keywords: Drone, UAV, Autonomous, Real-Time Object Detection, Radar, LiDAR, Jetson TX2, RPi4, Microcontroller, SSD-MobileNet V1, YOLOv3, RPLiDAR A2, Velodyne VLP-16, Ancortek SDR-980, NVIDIA, ROS

Executive Summary

UAV technology is used everywhere from military use to consumer use. Given that UAVs are unmanned, a shift towards adding artificial intelligence (AI) to these vehicles have increased with varying research and projects each tackling the project in a completely different way, with costs ranging up to thousands of dollars. While the task of autonomous navigation can be tackled through numerous approaches, this projects aims towards creating a stable real-time object system which can easily be mounted to these vehicles to create a universal baseline representing how UAVs understand their environment and allow further modifications to be implemented without reinventing the wheel.

From a technical standpoint, an object detection/semantic segmentation system can be crafted through visual, acoustic, radio frequency, or Radar/LiDAR/SONAR detection (Seidaliyeva et al., 2020, p.2). Therefore, the candidate solutions will examine previous attempts at creating this kind of system and will propose systems while keeping in mind the following constraints. Since this system will be targeted for UAVs, more specifically drones of varying sizes, the weight of added microcontrollers and components may impact flight ability (Seidaliyeva et al., 2020, p.4). Another aspect to keep in mind is short circuit prevention due to various water or weather hazards, so sourcing parts that can withstand these hazards are useful and will cut back on the cost to make the system impenetrable (Kentsch et. al, 2020, p. 1). Legalities may also impact the project as every country has varying regulations regarding UAV/Drone technology, as well as Radar bandwidths (de Haag et al., 2016, p. 4). The last constraint to keep in mind is overall cost since the project's goal is to create an affordable, implementable system.

The proposed solution involves utilizing the power of NVIDIA's Jetson microcontrollers, specifically the Jeton TX2 or Nano, paired with an RP LiDAR sensor. By pairing these components, a system based on visual CNN identification can be established by running some code on the microcontroller which is connected to the camera. The RP LiDAR yields a $12m$, $360°$ range around the UAV to avoid collisions and can process the data to the microcontroller for further analysis or autonomous navigation using ROS in conjugation to the camera (Sagarage et al., 2019, p. 6-7). Furthermore, the projected cost of the proposed solution estimates approximately $\$885$, and weighs about $436g$. According to their datasheets, the chosen components are all able to withstand weather hazards except for the microcontroller which would need to be embedded within a case. The solution also defeats radar legalities but runs into standard UAV laws.

With ever growing businesses such as Amazon or Google implementing drone delivery, as well as scientific research such as surveying Earth's ecosystems to combat climate change, UAV technology becomes a powerful tool for a variety of tasks. If the proposed system is successfully built, companies may work towards implementing the system and work towards modifying it for their specific tasks. Furthermore, scientific use may involve using the technology to navigate and record data in areas that could pose a threat to humans. Military use can involve a variety of tasks such as using the UAVs for maintenance on larger equipment, surveying landscapes, or delivering supplies.

Table of Contents

List of Figures

List of Equations

Problem analysis

This section will provide insight on real-time object detection for autonomous drones and Unmanned Ariel Vehicles (UAVs), stating an overview of the problem at hand, the fundamentals behind sourcing a solution, and discuss previous attempts and constraints to solve the issue.

Overview of the problem and its significance

As drone/UAV technology becomes ever evolving, industries ranging from the military to standard consumer delivery are relying on these vehicles for a variety of tasks. A study led by Ulzhalgas Seidaliyeva, discusses that one technological problem many developers are facing involves real-time detection for autonomous (AI controlled) drones/UAVs, which will allow these robotic vehicles to identify other vehicles, people, or necessary objects so they can process information and fulfill various tasks (2020, p. 1). UAV detection/identification is currently accomplished by various methods with ranging success. Seidaliyeva explains, “acoustic, visual, radio-frequency signal-based, and radar detection”, are all different ways varying studies have shown to operate an autonomous vehicle, which can be improved upon through developing a new system consisting of various sensors/cameras, software, and/or IMU data transmission (p. 2).

Eliminating the threat of possible collisions or security threats by developing a fast, general, stable object-detection/semantic segmentation AI, will not only improve upon consumer use, delivery industries, and military, but also play a role in various scientific breakthroughs such as space travel or identifying changes in the Earth’s forests and ecosystems to combat climate change (Kentsch et. al, 2020, p. 1).

Stem fundamentals

While UAV detection has a variety of methods to develop ranging from Acoustic detection to Radio-Frequency Signal-Based detection, the Radar or Visual/Convolutional Neural Networks (CNNs) methods have become the most defining solutions in object detection.

CNNs have existed since the 1980s as a primary method for machine learning, but according to Luo’s research, it was not until 2012 in which the AlexNet algorithm became the first CNN to establish image processing recognition/classification, which would later inspire CNNs such as MobileNet or YOLOv3 (Luo et al., 2020, p. 1). CNNs operate using a network of layers based upon three fundamentals: Convolutional, Pooling, and Normalization/ReLu. The layers consist of an input, output, and series of hidden layers that rely on element-wise matrix multiplication or the dot product. This can be demonstrated by the general block diagram below:

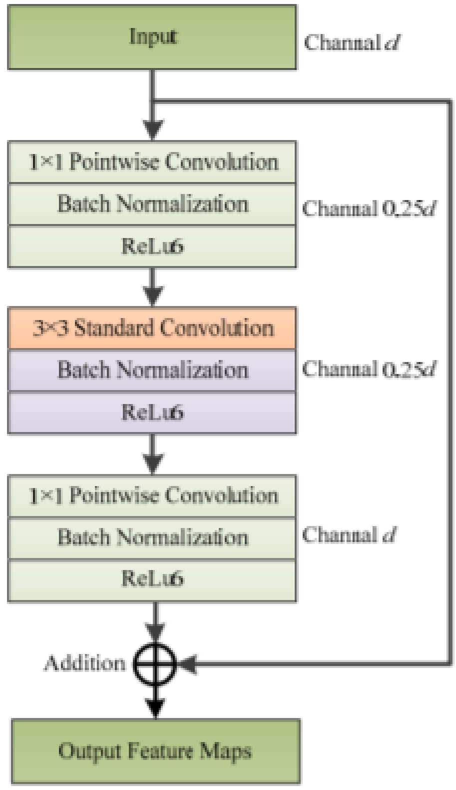


Figure 1. Basic structure of a CNN

Source: (Yu et al., 2020., p. 9)

Brandon Rohrer explains the concept that information is converted into a two-dimensional array where information such as an image, audio, or other data are converted into pixels represented by a "1" for a light pixel, or a "-1" for a dark pixel. After converting the information into pixels, the computer cross examines with a database of images it is already trained upon and then determines whether the information passed in matches (2016.). This process is known as filtering, filtering operates by lining up each pixel one by one and performing element-wise multiplication or the Hadamard product, as shown in the following example equation:

$$\begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix} \circ \begin{pmatrix} b_{11} & b_{12} & b_{13} \\ b_{21} & b_{22} & b_{23} \\ b_{31} & b_{32} & b_{33} \end{pmatrix} = \begin{pmatrix} a_{11}\,b_{11} & a_{12}\,b_{12} & a_{13}\,b_{13} \\ a_{21}\,b_{21} & a_{22}\,b_{22} & a_{23}\,b_{23} \\ a_{31}\,b_{31} & a_{32}\,b_{32} & a_{33}\,b_{33} \end{pmatrix}$$

Equation 1. Hadmard Product Example with Matricies *(1)*

Source: (Admin., 2019.)

The computer takes the sum of all pixels that are a match and divides by the total number of pixels in the image to determine the accuracy of the recognition. The final number is then updated to a new array to create a sort of "map" which the computer will determine the image from after the filtering process is completed. The process of filtering every pixel in the image is known as convolution. The convolution layer is entirely comprised of a stack of filtered images, which then get shifted for pooling. The pooling layer creates a window of the filtered images, usually 2 or 3 images, and applies the window across the convolution layer to find the maximum number that was stored in the array. This number gets stored in a new array and the process is repeated by shifting the window to examine the next set of images. After the maximum pooling image is found, this new "map" represents the previous information in a compressed set of data to prevent taking up too much memory. Finally, at the end of the pooling layer the "map" is normalized by changing any remaining negative value into "0", allowing the computer to determine what it is seeing and provide an accuracy (Rohrer. 2016). The process can be repeated multiple times, which is known as Deep Stacking. An example can visually be examined in the AlexNet architect below:

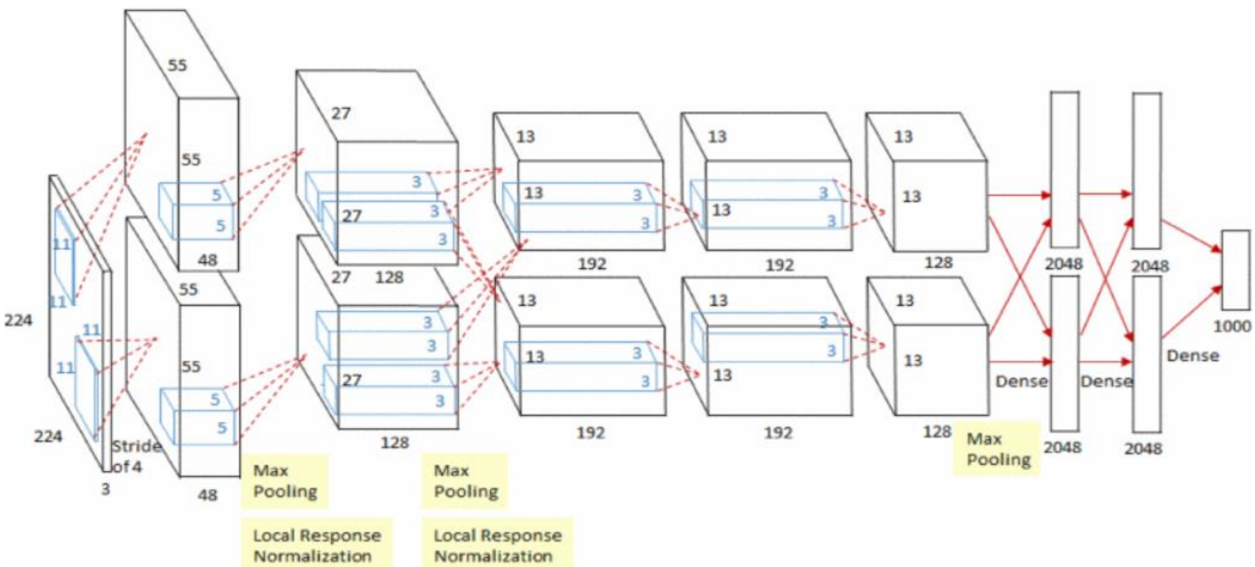


Figure 2. Example of a CNN, the Architecture of Alex Net

Source: (Zamri et. al., 2019 p. 2)

While using a CNN is an efficient approach for creating a strong object detection AI in robots, the other end of the spectrum, Radar detection, still holds its advantages against the algorithm. Ground Penetrating Radar Detection (GPRs) could be essential for scientific research and military use since a Radar detector will sense an object such as a landmine that may be visually hidden from a CNN to identify. A general block diagram for a Radar detection system is as follows:

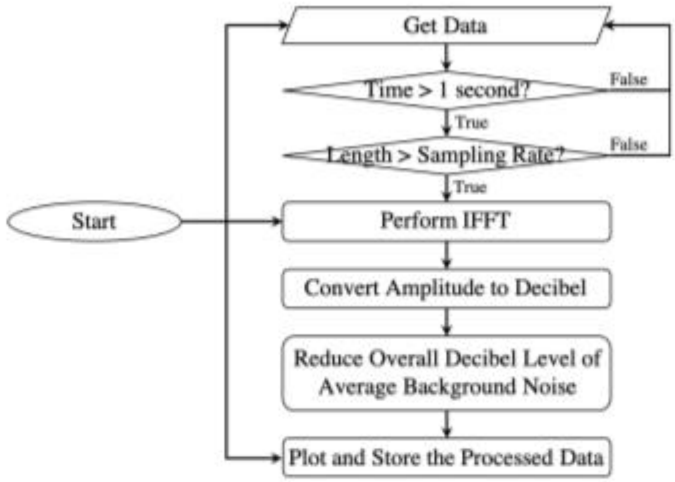


Figure 3. General Block Diagram of Radar Detection

Source: (Park, 2015., p. 8)

Šipoš and Gleich's study explains that GPRs in UAVs can work as follows, an Electromagnetic (EM) wave gets transmitted or received using an antenna. The EM waves reflect from metal and if a signal is received by the antenna, "then amplitude of the bounced signal is above the receivers noise floor, this change can be detected" (Šipoš & Gleich, 2020, p. 2). While using a Radar approach may detect objects more accurately than a visual approach such as the CNN method, creating a GPR may be more expensive and require more components contributing to the overall weight (Šipoš & Gleich 2020, p. 3-5):

A lighter weight, low-cost alternative lies based in technology such as LiDAR or SONAR which stems from concept of Radar. Methods using LiDAR or SONAR work in a similar manner as Radar, but their inputs are different waves from Radar's radio wave input. LiDAR feeds in light waves, while SONAR is based around sound. The choice between using Radar, LiDAR, or SONAR depends on the environment and purpose of the project. The National Ocean Service informs, LiDAR instruments typically consists of using a laser, a scanner, and a specialized GPS receiver to perform either a Topographical analysis, which uses a near-infrared laser to map land, or Bathymetric analysis, which is used to measure the seafloor (2012). To collect data, an airborne laser is aimed towards the ground and the beam of light is reflected by the surface. The sensor records the reflected light by measuring its range, which is then combined with IMU and GPS data to produce a, "point cloud", consisting of three-dimensional spatial coordinates being latitude, longitude, and height. (National Ocean Service. (2012).). Upon filtering, the LiDAR data can then create a Digital Elevation Map (DEM) at an estimated grid size determined by,

$$S = \sqrt{\frac{A}{n}} \tag{2}$$

Where S is the size of the DEM determined by A the area and n the number of points. Furthermore, the DEM may also be constructed by considering L the length of a transect and N the number of inflection points observed (Liu, (2008), p.11),

$$S = \frac{L}{2N_P} \tag{3}$$

A basic block diagram for LiDAR may appear as follows:

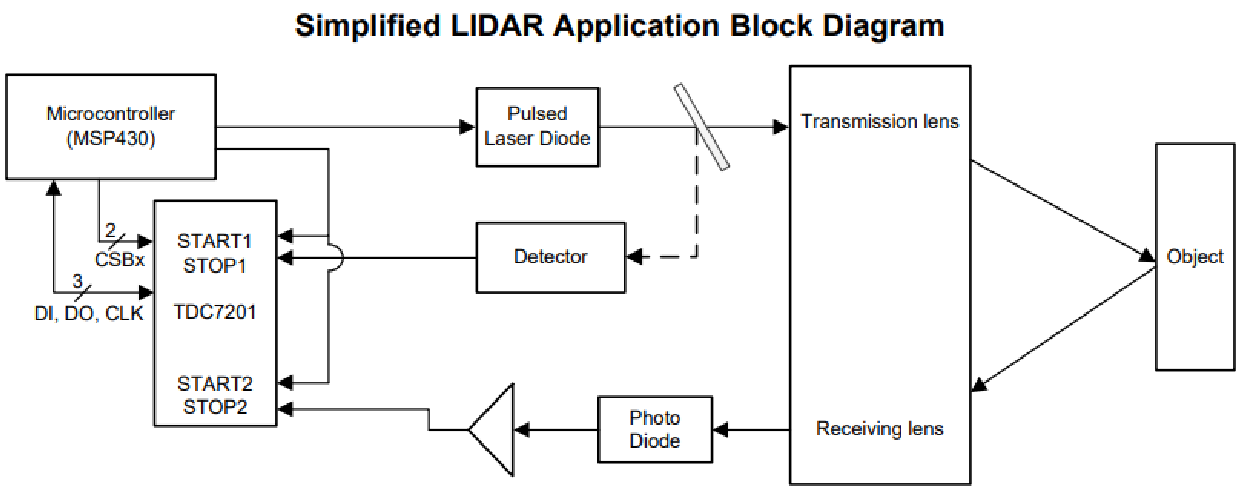


Figure 4. Block Diagram of a LiDAR Sensor

Source: (Peterson, 2020.)

Lessons from prior responses to the problem

When designing real-time object detection for a UAV, some considerations are worth noting based on previous studies. Utilizing the methods of both Radar and Visual classification, are why creating a multi detection method will be more beneficial and accurate than relying on one source of image detection.

Rohrer states because of the algorithm of a CNN the biggest limitation of using a CNNs is that CNNs focus primarily of "spatial" patterns recognized within data. If an image cannot be recognized based on patterns, a CNN will not detect the image as an object and the CNN will become useless (Rohrer. 2016). That being said, training a CNN will be the most important aspect of the process by feeding in various datasheets, since Luo's study states, "[Their tests using CNN classification] is based on the characteristics of vehicle targets, we think that targets with similar characteristics can also be detected by our proposed method", to combat any accuracy errors, multiple datasheets and parameters should be put into effect (Luo et al, 2020, p. 14)

In terms of Radar detection, Šipoš & Gleich found that one limitation of using Step Frequency Continuous Wave (SFCW) radar, "the acquisition speed. This is affected due to the switching between frequencies" (Šipoš & Gleich, 2020, p. 4). A different study conducted in 2015 using a Radar detection method based upon a Frequency-Modulated Continuous-Wave (FMCW) instead of SFCW, concluded that Radar detection may also need to combat noise due to the angle of the mounted Radar antennas and interfering signals such as Wi-Fi (Park et al., 2015, p. 15).

Creating a DEM is a great way to observe data using LiDAR, however, Liu cautions that DEMs are often oversampled in low relief areas and undersampled in high relief areas, so to combat this, data reduction should be considered, "one way to increase the details of the terrain representation is to increase the sample point density and decrease the grid size. This will lead to the redundancy of sample point and the increase of data size" (Liu, 2008, p. 11).

Project objectives and constraints

Objectives:

Objectives for this project are to improve upon previous attempts at creating an accurate autonomous drone by focusing primarily on the object detection aspect of the AI. Attempts to create a stable AI has included varying methods of approaching the problem. Implementing multiple methods to create a multimodal detection system based on CNNs and Radar/LiDAR Detection, can strengthen the overall system, however, it will require more components such as sensors and microcontrollers to be added (Seidaliyeva et. al, 2020, p. 3). Additionally, there are various paths to implementing both a Radar system and a machine learning algorithm. Rather than reinvent the wheel, it is probably best to modify an already existing CNN such as MobileNet, DenseNet, and YoloV3 (Chen et al., 2020, p. 4). In terms of Radar detection, using Radar or LiDAR will depend on the range, cost, and weight of implementing each method which will be explored in the candidate solutions section.

To create a vehicle with maximum accuracy, one goal will be to increase the range of detection so objects can be identified from a short or long trajectory. If a fast-stable real-time object detection is achieved by creating a multi detection system, perhaps this system could be implemented as a starting point for other robotics and autonomous vehicles, so other teams will not have to reinvent the wheel.

Constraints

By achieving the objectives above the following constraints must be considered when designing the AI and applying it to the overall drone for simulations:

- Adding a multi detection system may include adding new components and microcontrollers which will increase the overall weight and may impact the vehicles ability to fly (Seidaliyeva et al., 2020, p. 4).
- Considering how the object detection methods will be controlled and intertwined with the autonomous AI.
- Designing the detection to be accurate enough to withstand varying weather conditions and interfering signals (Kentsch et. al, 2020, p. 1).
- Designing a system that will be considered legal/will not require governmental approval (de Haag et al., 2016, p. 4).
- Considering an affordable and environmentally efficient solution for hardware implementation.

Candidate solutions

The following section of this paper creates a comparison and analysis of three possible solutions to advance autonomous object detection. Considerations among each of these solutions involve a comparison in design, cost, efficiency, and weight.

Scope of solutions considered

As Seidaliyeva's study states, the four primary modalities that can be used for drone detection/object classification are radar, radio frequency, acoustic sensors, and camera/visual sensors (Seidaliyeva et al., 2020, p.2). To ensure the maximum amount of accuracy, creating a multimodal system will aid the project in the long run, however, due to project constraints such as cost and weight, it is better to focus on integrating at least two of the four modalities. Given the scope of UAV technology ranges from military to consumer use, acoustic sensors, while low-cost, do not work as effectively in loud areas such as urban locations or areas with intense wind conditions. RF detection may be more energy efficient than radar detection, but many drones do not have RF transmission, so identifying other drones/UAVs may be a problem (Seidaliyeva et al., 2020, p. 2-4).

Combining Radar/LiDAR and Visual detection will allow the UAV to identify other aerospace objects, metals, weather formations, as well as people or custom targets, at both long and short range. The following sections will discuss various approaches to creating a cost-efficient, energy-efficient, light-weight multimodal drone by comparing various micro controllers, cameras, machine learning, and radar technology.

Explanation of candidate solutions

The factors of each solution rely primarily on the limitations of the chosen CNN and Radar/LiDAR technology, and any necessary components that may need to be added such as GPS, IMU, or additional microcontrollers.

RPi: SSD MobileNet V1 and Velodyne VLP-16 LiDAR

CNN

The SSD MobileNet V1 could be a potential choice for CNN machine learning. SSD MobileNet V1 was tested against four other CNNs in a study done at University of Tennessee Chattanooga (UTC). The SSD MobileNet V1 works similarly to the AlexNet discussed in the problem analysis section, the following diagram demonstrates the architecture of the network,

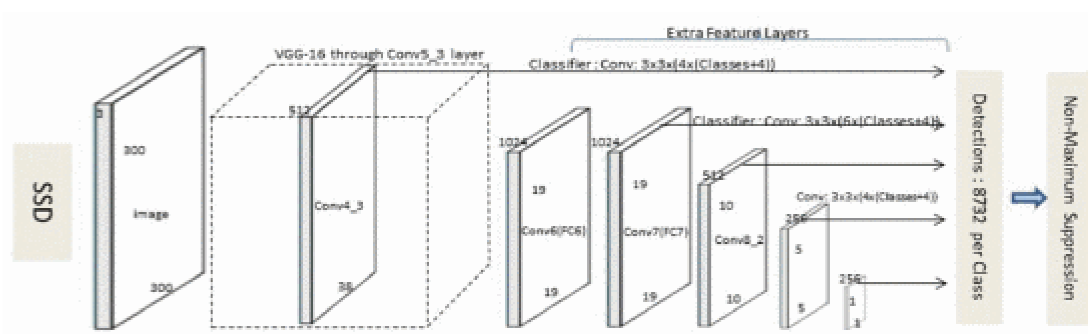


Figure 5. SSD MobileNet-V1 Architecture

Source: (Zhang et al., 2019, p. 120)

For the UTC study, the SSD MobileNet-V1 CNN was implemented using a RPi3 microcontroller, with the built-in camera kit, and showcased an average of 96% accuracy at 13 FPS among feeding in images of streets to identify cracks within (Qurishee, 2019, p. 43-47). After comparing CNNs upon still images, a live test was done by implementing the CNN into a drone for real time object detection. The results are shown in figure 7. While another CNN, Faster R-CNN Inception V2 delivered the highest accuracy, the framerate was trapped at 0.5 FPS in comparison to SSD MobileNet V1, which delivered the highest frames and accuracy (Qurishee, 2019, p. 53-54).

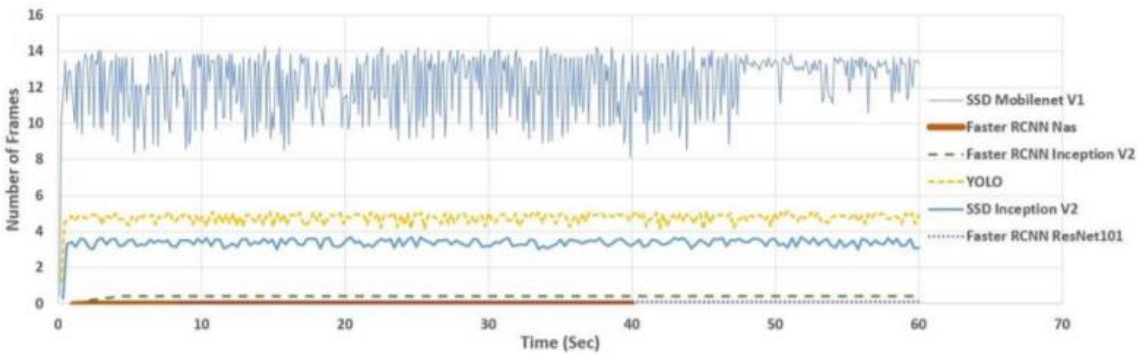


Figure 6. SSD MobileNet-V1 Accuracy results vs other CNNs

Source: (Qurishee, 2019, p. 54)

It should be noted that the study done at UTC involved short distance identification, the SSD MobileNet V1 was also tested against long distance satellite level classification in a study done by Istanbul Technical University (ITU). ITU found that YOLO, which UTC determined to be the worst CNN, to be the best at long distance for its speed and accuracy, and SSD MobileNet V1 was determined to be the worst in terms of accuracy at long distance. Faster RCNN was also examined to be fairly accurate and fast at long distance (Alganci et al., 2020, p. 18-20). The primary focus of this project will be short to medium ranged distance, so the SSD MobileNet V1 may be acceptable, by allowing the longer ranged identification to be picked up by the LiDAR implementation.

LIDAR

For speed and cost efficiency, a long-distance Topographical Airborne LiDAR sensor could be mounted on top of the drone to process information through MATLAB which will then be connected to the RPi using MATLAB's built in support and ROS nodes. A study led by Saied Pirasteh combined the use of LiDAR with R-CNN to determine change detection of urban landscapes. They created an algorithm shown using the following block diagram,

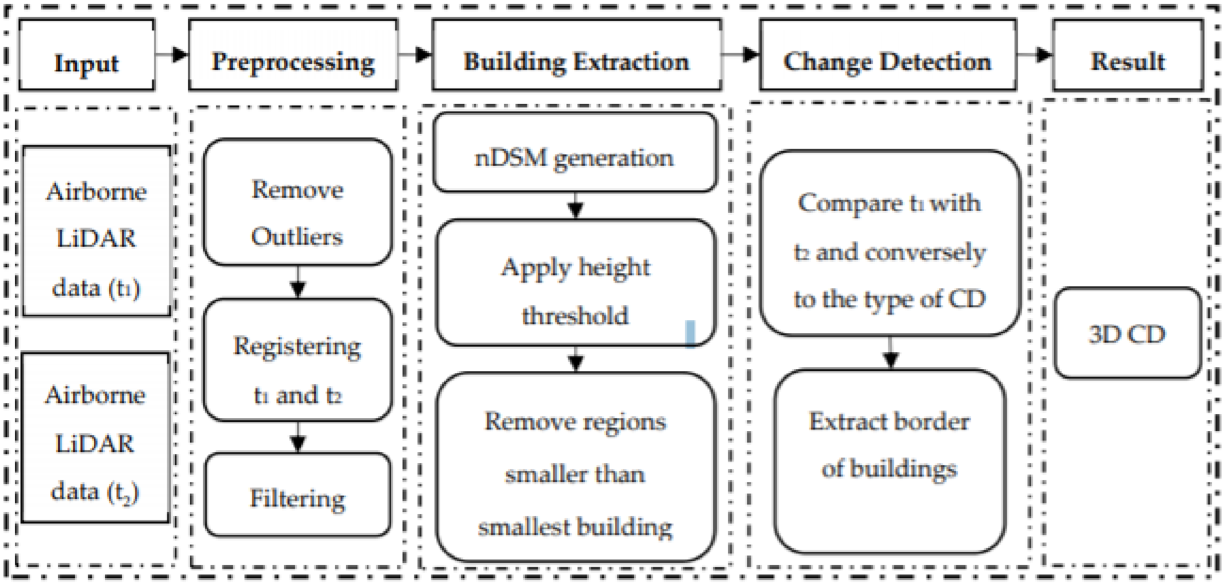


Figure 7. LiDAR and R-CNN Block Diagram

Source: (Pirasteh et al., 2019, p. 4)

By using a Slope and Progressive Window Threshold (SPWT), the team took a small window, slope, and large window threshold to distinguish the ground and non-ground points within the data (Pirasteh et al., 2019, p. 5). By taking the threshold, the Digital Elevation Model (DEM) and Digital Surface Models (DSMs) are created so the building extraction process of the block diagram can be ran by subtracting the the points within the raw scan that are not buildings, using the equation,

$$nDSM = DSM - DEM \tag{2}$$

Where nDSM is the new DSM that is ready for the change detection to be done in MATLAB (Pirasteh et al., 2019 p. 6). The initial LiDAR scan compared to the final DSM results are seen below as the program runs through the block diagram:

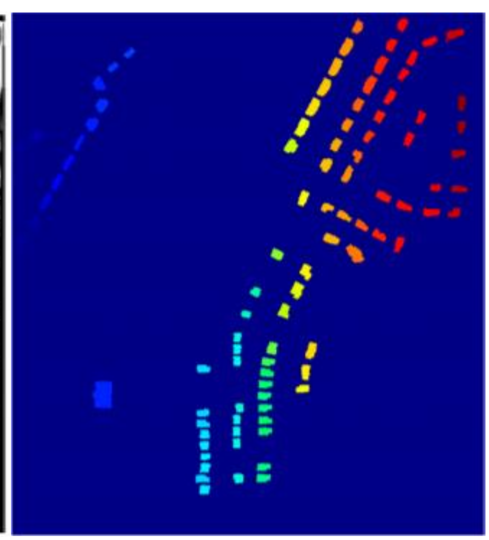
 

Figure 8. Input raw aerial scan Figure 9. Pre-Processing: Initial DSM of LiDAR Scan

Source: (Pirasteh et al., 2019, p. 9-10).

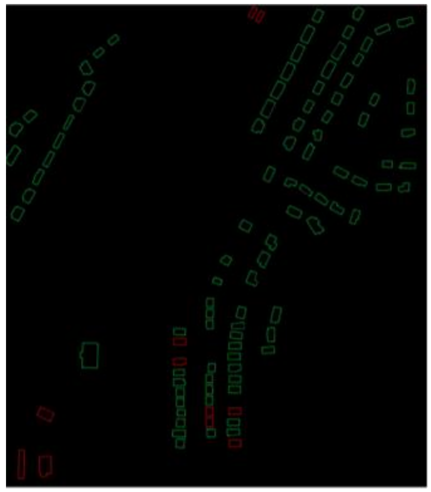


Figure 10. Building Extraction: LiDAR analysis after Change Detection. New Buildings are coded in red

Source: (Pirasteh et al., 2019, p.12).

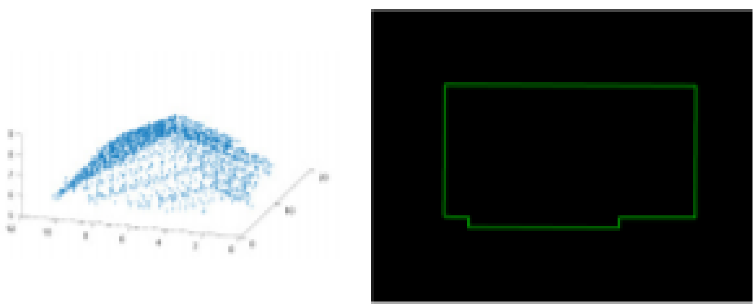


Figure 11. Change Detection: MATLAB Implementation using Point Clouds to transform LiDAR analysis into a 3D Model of one of the building's roof

Source: (Pirasteh et al., 2019, p.13).

Pirasteh also states that the LiDAR's data was more effective than using R-CNN because there is a 3D model that gets developed in comparison to a CNN's 2D scan, but using both together methods together are proved to be a successful method for change detection when tested using buildings. He concludes stating future work with this algorithm could be expanding the system to detect changes in trees and the ground. The bulk of work applying this solution to create a system with SSD MobileNet V1, would lie in building upon Pirasteh's algorithm using MATLAB to identify buildings, vehicles, trees, and landscapes. While the CNN is a different CNN than their study, SSD MobileNet V1 can be implemented by adding pre/post processing code and will be trained the same data as Pirasteh's study, as well as people, animals, and objects for short to medium range identification (Pirasteh et al., 2019, p.16).

Implementation

MATLAB can implement the CNN for short to medium range using the Deep Learning Toolbox, which can connect to the RPi and access SSD MobileNet V1 with the help of MATLAB's Support Package for Raspberry Pi Hardware (MATHWORKS, 2020, p. 2132-2135). Ideally, if the RPi is connected to the same router as the computer that is running MATLAB, a live script can be developed to view both the CNN object detection and LiDAR's DSMs. While Pirasteh's study never mentions the specific LiDAR sensor used, the same algorithm can be modified to use the raw scans of a Velodyne VLP-16 LiDAR sensor, used in a study done by Maarten Uijt de Haag upon comparing LiDAR to Radar. De Haag lists the specifications of the sensor, making it suitable based on range, "the VLP-16 consists of 16 separate laser rangers that cover a vertical field of view (FoV) of 300 (-150 to 150), have a measurement range of ~100m and a range accuracy of +/- 3cm, and rotate at rates up to 20Hz (i.e., a horizontal FoV of 3600)" (de Haag et al. (2016). p. 3). De Haag's study will be further referenced later for their use of a CWFM Radar sensor. Since this project is to create a stable object detection system for autonomous UAVs, a website could then be written using html or JavaScript and launched through the RPi to connect the rest of the drone's components by publishing and subscribing to ROS Nodes. This website would display the output of the MATLAB data as well as any GPS, IMU, or controllability. Downsides to this method include the total cost of the RPi microcontroller and Camera Kit, LiDAR sensor, having a stable Wifi connection, and relying on MATLAB which is a yearly subscription that may be fine for commercial work but not consumer use.

TX2: YOLOv3 and Ancortek SDR-980 CWFM Radar

CNN

While using an RPi is a low-cost, low-powered, microcontroller can be effective, the developers at NVIDIA have a variety of products that process information faster than the RPi. An analysis comparing the Jetson TX2 to RPi4 was conducted by Ahmet Ali Süzen at Isparta University. The Jetson TX2 although a larger board than any of the RPi's, contains a higher GPU leading to a higher performance. Süzen's study concluded that upon implementing various CNNs, the TX2's performance ran the best using SSD MobileNet V1, though YOLO ran the second fastest and yielded more accurate results within a reasonable time for person detection. Faster-RCNN and R-FCN were also compared and performed with the highest accuracy but ran slower than the other two (Süzen et al., 2020, p. 2). Given that the Radar implementation for this solution can detect up to $25m$ and the Jetson TX2 is compatible with most V4L2 USB cameras to be plugged in, YOLOv3 will be the chosen CNN for this solution given it's open source accessibility, real-time object detection and ability to reach up to 67 FPS with 81.5% accuracy on the NVIDIA TX2 (Xu et al. (2018). p. 1336). As Ren Ying explains in his study regarding airport and ship target detection YOLOv3, "uses a number of well-formed 3×3 and 1×1 convolutional layer and later uses multi-scale predictions to structure some residual networks. Finally, it has 53 convolutional layers and is called as Darknet-53" (Ying. (2020). Ying's study concluded by experimental results that YOLOv3 was able to collect and label satellite images on the TX2 at about $0.56s$ and report an accuracy rate of 91.48% for airports and 93.89% for ships. YOLOv3's accuracy algorithm works as follows,

$$P = \frac{T_p}{T_p + F_P} \tag{3}$$

Where $\mathrm{T_p}$ is the number of targets correctly detected, $\mathrm{F_p}$ is the number of objects missdetected (Ying, 2020.). YOLOv3 proves to be valid long-range CNN when implemented on the TX2, compared to UTC's study when ran on the RPi4. Visual results of Ying's study can be seen below,

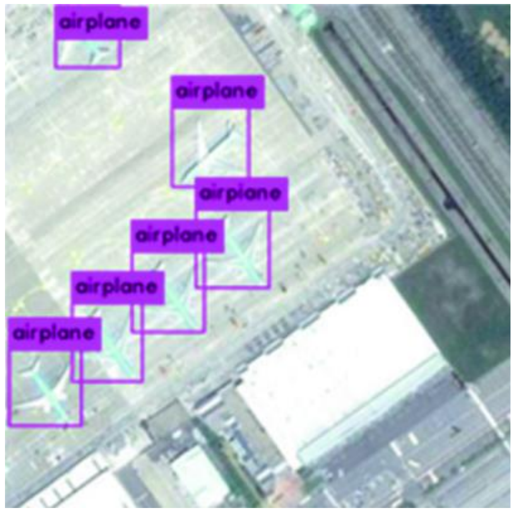


Figure 12. Visual Results of YOLOv3

Source: (Ying, 2020)

Given that YOLO has the power of possible long distance, the camera selected will be the ZED 2 Stereo Camera System by StereoLABS for its 120-degree wide-angle field of view and built in Barometer, Magnetometer, and IMU sensors. The choice of implementing a ZED 2 comes from a study done by Abhimanyu Chadha in which he used a TX2, ZED camera, and LiDAR to create a vision-based localization system for a GPS denied environment (Chadha, 2020, p. 27). Chadha's study used the TX2's CUDA processor cores to divide power between YOLOv3 and ZED 2, which in turn initially led to a suffering FPS until they reduced the detection quality from 2K to 720p. While this solution does not use LiDAR, it should be noted Chadha's team besides applying the ZED to YOLOv3, used the ZED in conjugation with the same LiDAR system proposed in the first solution to create their environment (2020, p. 27). He also noted the ZED's compatibility with various deep learning networks leaving future design improvements to not only work with versatility of YOLOv3, but to be mounted on a UAV and swapped with other CNNs depending on the environment or specific classification/detection task if one CNN has advantages over another (Chadha, 2020, p. 49).

Figure 13. Chadha's visual results using the ZED 2 directly on the TX2]

Source: (Chadha, 2020, p. 37)

Radar

An advantage of implementing a low cost, light weight, Continuous Wave Frequency Modulated (CWFM) Radar sensor is it will detect the presence of rain, fog, mist, snow, and dust. Unlike a LiDAR sensor, Radar detects at a larger wavelength, producing a wider, but less accurate, range (de Haag et al, 2016, p. 3). A study done at University of Pennsylvania merging LiDAR and Radar solutions, chose to modify a Ancortek SDR-980 Radar sensor to their selected frequencies and implement the system by plugging in the processor into a PC via USB compatible with the Radar's graphical user interface (GUI) on Windows. A block diagram for the Radar module is as follows,

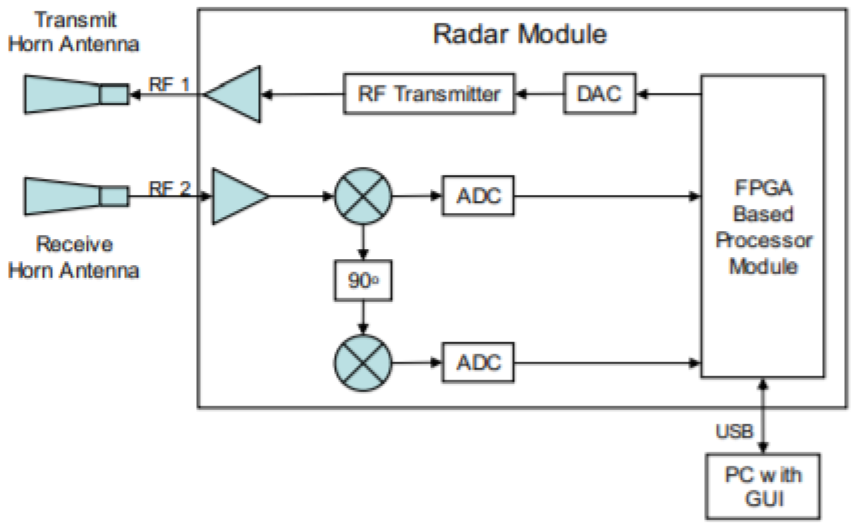


Figure 14. Ancortek SFR-980 Block Diagram

Source: (de Haag et al., 2016, p. 5)

Upon a full-flight test, a baseline radar performance test is done to calibrate the Radar sensor and Radar cross section enhanced X8 target platform. The team then collected data in real-time by operating at their selected frequency of $9.400GHz$ in a CWFM mode over a $200MHz$ bandwidth, collecting 1024 samples for every other $1ms$ sweep, totaling $180s$ worth of data per trial (de Haag et al., 2016, p. 10). The Radar's data could then be post-processed using a language such as MATLAB or Python for further analysis. Using MATLAB, the team created range vs velocity and range vs time in-and-out range profiles for both full-flight tests and Doppler cross-over tests. Figures 15-18 demonstrate what the Radar sees by measuring the ranges of how far the Radar sensor can detect with respect to time and velocity (de Haag et al., 2016, p. 10).:

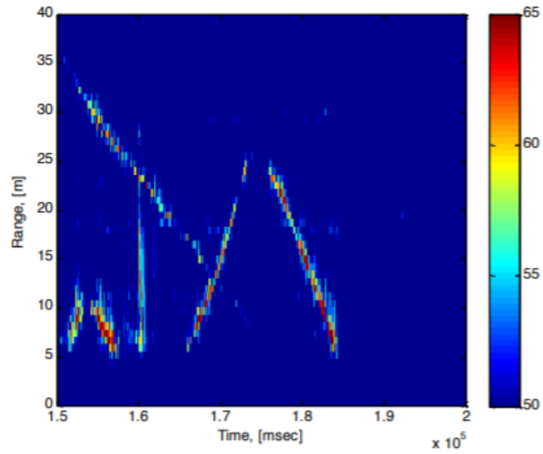


Figure 15. In and out of Range, Full-Flight Radar Performance, Range vs Time

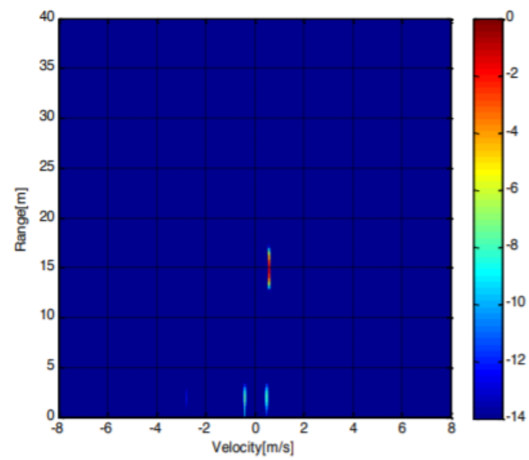


Figure 16. In and out of Range, Full-Flight Radar Performance, Range vs Velocity

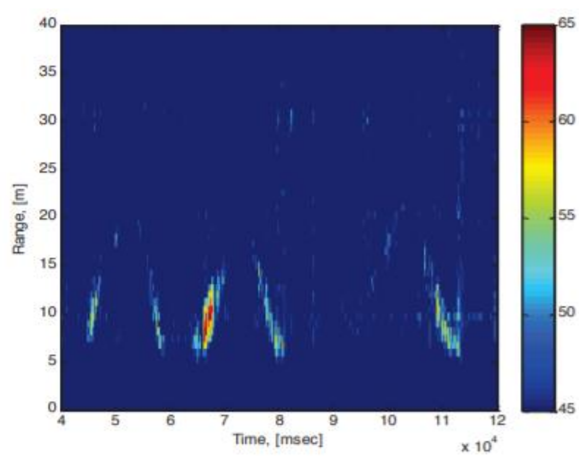


Figure 17. In and Out of Range, Doppler Crossover, Range vs Time

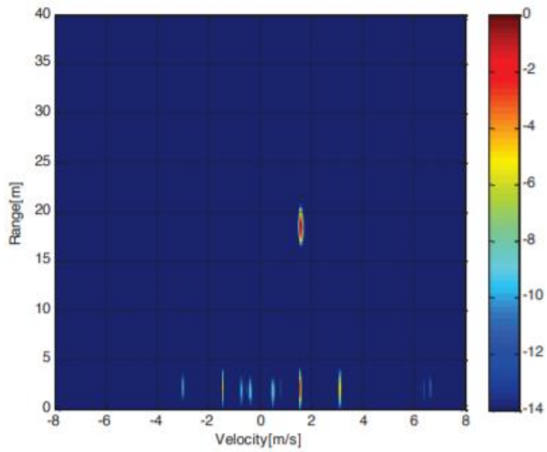


Figure 18. In and Out of Range, Doppler Crossover, Range vs Velocity

From these tests, De Haag's study concluded the Ancortek SDR-980 was able to detect both Phantom and RCS e-x8 platforms out of range approximately at $25m$, averaging a velocity in range of $\pm 4\,{}^{m}/_{s}$ at $0.8\,ms$. At a closer range, "less than $5m$ clutter was observed and a target void was observed as the target sUAS passed through the zero Doppler crossover point, which can occur for a compact CWFM installation" (de Haag et al., 2016, p. 10).

Implementation

Given that this solution keeps the CNN and Radar systems separate, the Radar data could be mapped using Python's Jupyter notebooks instead of MATLAB's Live Script, since it is open source, easy to use, and there are plenty of data analysis libraries that create models similarly. Given that the TX2 can run Ubuntu and NVIDIA offers an entire wiki through their website on merging Python, C++, and ROS, any scripts written can be generated on the TX2 to take in the data from the Radar's Windows GUI and map the detection. One possible disadvantage should be noted that implementing an airborne radar sensor will require obtaining an appropriate radar frequency band and approval to operate at the selected frequencies by the Federal Communications Commission (FCC) (de Haag et al., 2016, p. 4).

TX2: SSD MobileNet V1 and RPLiDAR A2

CNN

The SSD MobileNet-V1 will be the chosen CNN for this solution given because of the same reasons it was selected for the first solution, such as the real-time object detection and the ability to reach up to 46 FPS with 84.2% accuracy on the NVIDIA TX2 (Xu et al., 2018, p. 1336). Furthermore, NVIDIA offers full support of using any version of the SSD MobileNet CNNs through the official Jetson Inference Guide, which consists of an open source GitHub guide, which walks the user through training CNNs with pretrained or custom models using Pytorch or TensorFlow/TensorRT to optimize the model. The GitHub also incorporates a C++ or Python API for real-time detection, image classification, and segmantic segmentation, through the API's dectectNet, imageNet, or segNet, each offering the user to choose from a wide range of CNNs to train off of including the SSD MobileNet V1-V3, as seen below (NVIDIA, 2019, July 23.).

Network	CLI argument	NetworkType enum	Object classes
SSD-Mobilenet-v1	`ssd-mobilenet-v1`	`SSD_MOBILENET_V1`	91 (COCO classes)
SSD-Mobilenet-v2	`ssd-mobilenet-v2`	`SSD_MOBILENET_V2`	91 (COCO classes)
SSD-Inception-v2	`ssd-inception-v2`	`SSD_INCEPTION_V2`	91 (COCO classes)
DetectNet-COCO-Dog	`coco-dog`	`COCO_DOG`	dogs
DetectNet-COCO-Bottle	`coco-bottle`	`COCO_BOTTLE`	bottles
DetectNet-COCO-Chair	`coco-chair`	`COCO_CHAIR`	chairs
DetectNet-COCO-Airplane	`coco-airplane`	`COCO_AIRPLANE`	airplanes
ped-100	`pednet`	`PEDNET`	pedestrians
multiped-500	`multiped`	`PEDNET_MULTI`	pedestrians, luggage
facenet-120	`facenet`	`FACENET`	faces

Figure 19. Compatible CNNs with Jetson Inference

Source: (Franklin, 2020.)

LIDAR

In conjugation with selecting a CNN for machine learning, pairing a LiDAR sensor would allow for the system to create an accurate dynamic two-dimensional map of the surrounding environment. Yuthika Sagarage led a study in 2019 regarding LiDAR detection with the Jetson TX2, using an unbranded 1080p USB camera linked to a deep CNN, which acts as a classifier network for determining if the path ahead of the autonomous robot is blocked or not. The selected LiDAR was a 360-degree 2D RPLIDAR A2, which acts as a 2D point cloud to get spatial awareness of the propeller plane of a drone as well as the downward facing distance to maintain the accuracy of the altitude (Sagarage et al., 2019, p. 6-7). ROS's compatibility on the TX2, is used as a sort of middleware for publishing and subscribing the real time flight control system, in which they used the MAVROS package to issue Mavlink protocol commands from the TX2. The team represents their system by the following block diagram with the camera's feed and 2D LiDAR as inputs,

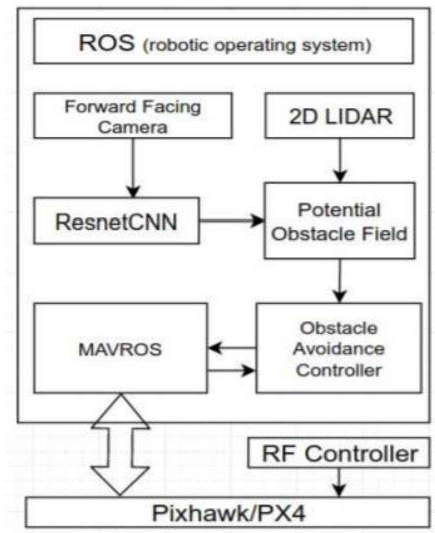


Figure 20. Block Diagram of Sagarage's Autonomous CNN and LiDAR system

Source: (Sagarage et al., 2019, p. 7)

The RPLiDAR A2 contains a range of $12m$ and builds a global map consisting of a histogram that is used to decide the maximum speed of the drone.

Implementation

Given that the TX2 could operate using ROS, Sagarage's study can be modified by implementing the detectNet API with SSD MobileNet V1, instead of using Res-Net, which according to their results resulted in a low framerate because they did not have TensorRT to optimize their models, which Jetson-Inference's API's already have built in (Sagarage et al., 2019, p. 10). While their study focused on mapping an environment for autonomous navigation and detecting obstacles directly in front of the drone, their basic framework already maps the 2D LiDAR and by running Jetson Inference's API in another ubuntu terminal, any object detection can be added to the preexisting framework and executed by creating ROS nodes. Unlike the other solutions, this allows everything to be done within the TX2 instead of outputting data to either MATLAB or a website, ideally, as long as the user is able to SSH into the TX2, everything can be controlled remotely, however, a website or GUI interface may be created to ensure consumer user friendliness. The camera in this solution could be any V4L2 USB camera compatible with the TX2, so upon researching through the TX2 wiki, the STEEReoCAM- 2MP Stereo Camera for NVIDIA Jetson Nano/AGX Xavier/TX2 will be selected for its ROS support, resolution of 1600x1300 with 30FPS operating at $3.3V$ +/- 5% at a fraction cheaper than the ZED 2 camera selected for the second proposed solution (E-Con Systems, 2019.).

Comparative assessment of candidate solutions

Table 1. Comparative Assessment of Candidate Solutions

	SSD MobileNet V1 and Velodyne VLP-16	**YOLOv3 and Ancortek SDR-980**	**SSD MobileNet V1 and RPLiDAR A2**
MicroControllers/Added Components Specifications: [Cost, Voltages, Power, Current, Weight]	RPi 4 $35 15W +5V 2A 42g	NVIDIA Jetson TX2 Developer kit $450 15W +12V 5A 85g	NVIDIA Jetson TX2 Developer kit $450 15W +12V 5A 85g
Camera Specifications: [Cost, Voltage, Weight]	RPi High Quality Camera $50 3.3V 3g	Any V4L2 could be selected, but the ZED 2 will be chosen based on Chadha's study. Note: some quality of the camera may be diminished to compensate FPS $449 5V USB powered 124g	Any V4L2 could be selected, the STEEReoCAM - 2MP Stereo Camera for NVIDIA Jetson TX2 was selected based upon the Jetson TX2 Wiki. $99 3.3V +/- 5% USB powered 161g
LiDAR or Radar [Cost, Weight]	LiDAR Velodyne VLP-16 $4000 600g	Radar Ancortek SDR-980 (Uncertain about the price, an email would need to be sent to contact about prices) FCC Approval makes this solution less consumer friendly.	LiDAR 2D RPLIDAR A2 $335.44 190g
Eco Friendly?	Yes	Yes	Yes
Controllability	Requires Wifi for website and MATLAB. A router may have to be mounted to connect the website and ROS to the person controlling the Drone Matlab subscription	Everything can be done in TX2 through SSH or using the Radar's GUI on a Windows Laptop/PC. If the PC is connected to the same network as the TX2, it can be controlled.	Using NVIDIA's Jetson Inference API support, SSD MobileNet V1 and the Camera can be controlled.

	$860 per year, making this less consumer friendly		
Weather Hazards?	While the RPi Camera with CNN may suffer, the LiDAR can withstand weather	Radar can detect through varying weather hazards	LiDAR can withstand Weather hazards
Could Autonomous Navigation be implemented in the future?	Yes	Yes	It already stems from an autonomous navigation system
Total Weight	≈645g	≈209g	≈436g
Total Cost	≈$4,945	≈$899	≈$884.44

Project recommendations

The following section will interpret information from the candidate solutions to determine which solution will be most efficient in satisfying the issue at hand. Furthermore, any design and implementation challenges for this solution will be discussed, as well as the impact the system will have on commercial and consumer use.

Proposed solution

Pairing the SSD MobileNet V1 and RPLiDAR A2 proves to be the most efficient solution for a plethora of reasons when compared to the other candidate solutions. Table 1., the comparative assessment of candidate solutions, lays out the specifications of each solution that would be needed to achieve the goals derived from the objectives and constraints section. Considering this system is meant to be implemented for a general audience of consumer or commercial use, the cost of each solution plays a large factor in determining which solution could be the best. The third solution consisting of pairing the SSD MobileNet V1 and RPLiDAR A2 ranks best coming in at approximately $\$884$, while the first solution which also uses the SSD MobileNet V1 but with Velodyne VLP-16 weighs in as the most expensive solution of approximately $\$5k$ due to the cost of a MATLAB subscription and the Velodyne, which is probably unreasonable to be a consumer product since most consumer drones typically are less than $\$2k$. Radar implementation with a CNN such as second solution with YOLOv3 and Ancortek SDR-980 comes close to the price range of the SSD MobileNet V1 and RPLiDAR at approximately $\$900$, however this is without factoring in the cost of the radar solution which Ancortek does not blatantly say unless contacted upon.

Each solution consists of two factors, the CNN and the Radar/LiDAR detection. There are only two CNNs to compare since SSD MobileNet V1 is used in two solutions versus YOLOv3. Alganci's study expressed the advantages of using the SSD MobileNet V1 or YOLOv3 due to their accuracy, with SSD MobileNet V1 consisting of the worst accuracy at long distance and YOLOv3 ranking the best for long distance (Alganci et al., 2020, p. 18-20). When examining the results of the SSD MobileNet V1 at a short to medium range on the TX2 vs the RPi, the CNN tends to run faster but slightly less accurate at 46 FPS and 84.2% accuracy on the TX2 (Xu, et al., 2018, p. 1336). The RPi results in 13 FPS with 96% accuracy according to Quiresh's study (2019, p. 47). Using a solution regarding the SSD MobileNet V1 would allow for a faster efficient solution at smaller ranges than the YOLOv3, which is more applicable for modern consumer and commercial drone use and further range amplification could be achieved by upgrading the camera for a better range (Qurishee, 2019, p. 53-54). From this aspect, selecting the SSD MobileNet V1 paired with RPLiDAR seems the most reasonable for its speed and better camera, although the accuracy does suffer slightly and using a TX2 will require more power than running an RPi.

The choice between choosing a solution containing Radar versus a solution with LiDAR sensing is based upon a matter of specifications. An ethical choice plays a factor as well, since this is a general solution and pushing this solution out for consumers should allow them to understand the technology easily. Implementing a Radar system could be confusing for a typical consumer, based on the graphical analysis of the Doppler Cross tests, furthermore, a simple executable program would have to be generated before placing the system on the market because it is not clear how many consumers know how to code. A language such as Python would be chosen for this solution because of its open source licensing and data analysis libraries, which in a 2020 analysis from statista examines that only 44.1% of programmers are currently using python (2020). The benefits of using LiDAR over Radar also stems from FCC approval to operate the Radar sensor, furthermore, the only approval of using the other two solutions would be owning a drone license (de Haag et al., 2016, p. 4). Choosing the LiDAR solutions could potentially be more user-friendly due to their ROS implementation, since ROS could be accessed through GUI interfaces such as Gazebo or RVIZ. One of the most technologically advanced LiDAR sensors is the Velodyne VLP-16 3D LiDAR which has a range of approximately $100m$, rotating up to $20Hz$ (de Haag et al., 2016, p. 3). The downside to the Velodyne VLP-16 lies with a cost of $\$4k$ and a weight of $600g$. The RPLiDAR A2 on the other hand is a much more light-weight, affordable sensor at $\$335$, weighing $190g$, however, this sensor is a short to medium distance LiDAR detection with a range of $12m$, mapping in 2D instead of 3D (Sagarage et al., 2019, p. 7).

The problem at hand primarily discusses creating an easy, accurate detection system for autonomous drones/UAVs. Pairing the SSD MobileNet-V1 with an RPLiDAR sensor is the best route for its affordability, simplicity, and weight. While this solution would require more power than using a solution such as the SSD MobileNet-V1 and Velodyne VLP-16, this solution uses a better camera for accuracy, does not necessarily require wifi, nor MATLAB, and already stems from research that further implemented autonomous navigation systems. While the range of the RPLiDAR suffers compared the Velodyne, this should not matter because a range of $12m$ is more than enough for a drone to understand what it is about to collide with, and the camera in conjugation with SSD could detect further ranges in front of the drone. Another downside to using the RPLiDAR A2 is that this is a 2D LiDAR in comparison to Velodyne's 3D LiDAR, however, it is worth noting since Sagarage's study, there is now an RPLiDAR A3 which is a 3D LiDAR sensor the system could upgrade to use. Using the RPLiDAR A3 would act similarly to the RPLiDAR A2, except this sensor costs about $\$600$, ranging at $25m$ like the Velodyne, and weighing $190g$ (Huang, 2020.).

Design and implementation challenges

The creation of a reproducible fast-stable real time object detection system could be a foundation for future Drone/UAV technology improvements. The proposed solution could be a step towards the right direction if the design and implementation challenges faced are overcame. In terms of the mechanical aspect of the project the selected Jetson TX2 microcontroller, while it is critically acclaimed for its fast GPU and efficiency, it is a larger and heavier microcontroller that would cause issues depending on the drone's size and how much weight it can hold. A few work arounds could solve this issue, one being that this system would simply have a requirement for larger, more durabile drones. The other option is that the TX2 could be replaced with the smaller, less effiecient NVIDIA Jetson Nano, however this may lead to a lack in performance issues since the GPU is slightly downgraded and contains a smaller memory (Süzen et al., 2020, p.2). However, given this idea two systems could be offered, one based upon the TX2 and one based upon the Nano. As stated above in the proposed solution's section, the RPLiDAR A2 should be upgraded to the RPLiDAR A3, which much like the microcontroller would have to be mounted as well, so keeping in mind the weight, range, and mounting location of the RPLiDAR A3 would be something to consider (Huang, 2020.). The last component to be mounted would be the camera, which in the proposed solution was the STEEReoCAM, but as stated previously, if this system were to be implemented on an already existing Drone/UAV, a USB V4L2 compatible camera could be plugged right into the TX2 or Nano and swapped with the original drone camera or mounted on to fit the individual drone's usage.

Another aspect of designing this system includes understanding the electrical engineering properties. To make this system function, an overall block diagram should be created to understand the workflow of the system because there will need to be some additional components needed. Looking back at the comparative assessment table the TX2 requires at least 15W of power, but additionally an external hard drive needs to be added via SATA adapter to give the TX2 memory (Süzen et al., 2020, p.2). Furthermore, a power supply should be implemented with consideration of surge protection from shorts, as well as creating a voltage divider so the camera and lidar does not take in too much power. While the camera and LiDAR detection can withstand varying weather hazards, the microcontroller needs to be placed within a case/electrical box so that nothing will destroy or short any components. In the case of strong weather conditions, a router to boost the connectivity of the microcontroller could be added but is not necessary since that primarily depends on the drone operators' use for the drone.

Anticipated project outcomes and impacts

Creating a stable real-time object detection system for drone/UAV technology could bring breakthroughs within scientific research, military use, and businesses. Combating climate change is an impending issue that could benefit from modifying this system for scientific research. An example of this already exists in Japan when a research team led by Sarah Kentsch used a UAV with object detection to evaluate the status and changes within a Japanese forests' ecosystem. They created a system trained on two datasets for winter and coastal forests and used the UAV to identify changes within plant/tree life and identify any invasive species that may impact vegetation life, additionally, they also incorporated semantic segmentation to observe changes in watersheds for further analysis (Kentsch et al, 2020, p. 4-10). By creating an accessible system based upon the proposed solution, future studies could carry out similar experiments within other locations as well as possibly having a role in creating an atmosphere on Mars.

While the military uses man operated drones for target detection, which raises a matter of ethical controversies, the proposed solution could still benefit the military humanely in terms of maintenance. Temple University currently is developing various UAVs/UUV technology under the direction of Dr. Li Bai, which will be sync'd up autonomously for a goal to identify corrosion and buildup on the sides of navy ships. If commercialized, the proposed solution could minimize the amount of research needed because of its accessibility that could be modified to train against buildup and corrosion. Furthermore, the navy has also led some projects by creating Unmanned Air Systems (UAS) using object detection to identify macro liter on sandy beaches, by implementing the proposed solution, further developments can modify a UAV and other autonomous vehicles to start to clean the liter after a UAS detects the liter and feeds the data to these vehicles (Goncalves et al., 2020, p.1).

Commercializing the proposed system could lead to potential businesses or consumers use. Companies such as Amazon and Google have developed drones to autonomously deliver packages, but a vast room for improvement lies within speed, payload, and autonomous navigation system (Shavarani et al., 2018, p.1). Developing the proposed system could potentially aid the drones' navigation system by giving them references to avoid with real-time object detection, in conjugation with the GPS navigation system they use to maneuver. Furthermore, manufacturing/inventory jobs may find the system to be useful when surveying equipment or for inventory checks. Depending on government regulation, consumer use may vary but would exist for the same reasons a consumer would want an autonomous drone, as well other autonomous detection robots such as the Roomba.

References

Admin. (2019). An introduction to hadamard product - deep learning tutorial. Retrieved September 13, 2020, from https://www.tutorialexample.com/an-introduction-to-hadamard-product-deep-learning-tutorial/

Alganci, U., Soydas, M., Sertel, E. (2020). Comparative research on deep learning approaches for airplane detection from very high-resolution satellite images. *Remote Sens., 12*, 458.

Chadha, A. (2020). Vision based localization of drones in a gps denied environment. Retrieved October 02, 2020, from https://vtechworks.lib.vt.edu/handle/10919/99887

Chen, L., Ding, Q., Zou, Q., Chen, Z., Li, L. (2020). Denselightnet: a light-weight vehicle detection network for autonomous driving. *IEEE Transactions on Industrial Electronics*, *67*, 12, pp. 10600-10609. https://www.aminer.cn/profile/long-chen/562c675945cedb3398c1187b de Haag, M. U., Bartone, C. G., Braasch, M. S. (2016). Flight-test evaluation of small form-factor LiDAR and radar sensors for sUAS detect-and-avoid applications, *2016 IEEE/AIAA 35th Digital Avionics Systems Conference (DASC), Sacramento, CA*, pp. 1-11, https://www.semanticscholar.org/paper/Flight-test-evaluation-of-small-form-factor-LiDAR-Haag-Bartone/31c607d1db1f1dd87f8e6865f11cfefcfc7b920f

E-Con Systems. (2019). STEEReoCAM- 2mp stereo camera for nvidia jetson nano/agx xavier/tx2. Retrieved October 02, 2020, from https://www.e-consystems.com/nvidia-cameras/jetson-agx-xavier-cameras/stereo-camera.asp?utm_source=GoogleShopping

Franklin, D. (2020). Jetson-inference github. Retrieved October 01, 2020, from https://github.com/dusty-nv/jetson-inference

Gonçalves, G., Andriolo, U., Gonçalves, L., Sobral, P., Bessa, F. (2020). Quantifying Marine Macro Litter Abundance on a Sandy Beach Using Unmanned Aerial Systems and Object-Oriented Machine Learning Methods. Remote Sensing, 12(16), 2599. https://www.mdpi.com/2072-4292/12/16/2599

Hayton, J. N. C., Barros, T., Premebida, C., Coombes, M. J., Nunes, U. J.(2020). CNN-based human detection using a 3d lidar onboard a UAV. https://home.isr.uc.pt/~cpremebida/files_cp/CNN_UAV_Lidar.pdf

Huang, T. (2020). RPLIDAR-A3 laser range scanner_ robot laser range scanner | SLAMTEC. Retrieved 22 October 2020, from https://www.slamtec.com/en/Lidar/A3

Jain, S. (2018). How to add Person Tracking to a Drone using Deep Learning and NanoNets. Retrieved 17 November 2020, from https://nanonets.com/blog/how-to-add-person-tracking-to-a-drone-using-deep-learning-and-nanonets/

Kentsch, S., Lopez Caceres, M.L., Serrano, D., Roure, F., Diez, Y. (2020). Computer vision and deep learning techniques for the analysis of drone-acquired forest images, a transfer learning study. *Remote Sensing*, *12*(8), 1287. https://www.mdpi.com/2072-4292/12/8/1287

Liu, X. (2008). Airborne lidar for dem generation: some critical issues. *Progress in Physical Geography: Earth and Environment, 32*(1), 31–49. https://journals.sagepub.com/doi/10.1177/0309133308089496

Luo, X., Tian, X., Zhang, H., Hou, W., Leng, G., Xu, W., Jia, H., He, X., Wang, M., Zhang, J. (2020). Fast automatic vehicle detection in UAV images using convolutional neural networks. *Remote Sensing*, *12*(12), 1994. https://www.mdpi.com/2072-4292/12/12/1994

National Ocean Service. (2012). What is lidar. https://oceanservice.noaa.gov/facts/lidar.html

NVIDIA. (2019, July 23). Two days to a demo. Retrieved October 01, 2020, from

https://developer.nvidia.com/embedded/twodaystoademo
MATHWORKS. (2020). Deep learning toolbox user's guide. https://www.mathworks.com/help/pdf_doc/deeplearning/nnet_ug.pdf

Park, S., Kim, Y., Lee, K., Smith, A.H., Dietz, J.E., Matson, E.T. (2015). Accessible real-time surveillance radar system for object detection. *Sensors*, *20*, 2215. https://www.mdpi.com/1424-8220/20/8/2215

Peterson, Z. (2020). Lidar system components for autonomous vehicles. https://octopart.com/blog/archives/2019/11/lidar-system-components-for-autonomous-vehicles

Rohrer, B. (2016). How convolutional neural networks (CNNs) work [Video File]. *End-to-End Machine Learning*. Retrieved from https://end-to-end-machine-learning.teachable.com/courses/how-deep-neural-networks-work/lectures/9533964

Pirasteh, S., Rashidi, P., Rastiveis, H., Huang, S., Zhu, Q., Liu, G., Li, Y., Li, J., Seydipour, E. (2019). Developing an algorithm for buildings extraction and determining changes from airborne lidar, and comparing with R-CNN method from drone images. *Remote Sensing*, *11*(11), 1272. https://www.mdpi.com/2072-4292/11/11/1272

Qurishee, M. A., Weidong, W., Owino, J., Ignatius, F., Mbakisya, O. A., Liang, Y. (2019). Low-cost deep learning UAV and raspberry pi solution to real time pavement condition assessment. *University of Tennessee Chattanooga.* https://scholar.utc.edu/cgi/viewcontent.cgi?article=1752&context=theses

Sagarage, Y., Jayawardena, J. K., & Kuruppu, M. (2019). Deep convolution neural network (CNN) using a monocular camera and a 2d lidar for robust quadrotor obstacle avoidance. *IET-Sir Lanka Network, Annual Conference 2019.* http://theiet.lk/26th-annual-technical-conference-research-papers-2/

Seidaliyeva, U., Akhmetov, D., Ilipbayeva, L., Matson, E. T. (2020). Real-time and accurate drone detection in a video with a static background. Sensors, *20* (14). https://www.mdpi.com/1424-8220/20/14/3856

Shavarani, S.M., Nejad, M.G., Rismanchian, F. (2018). Application of hierarchical facility location problem for optimization of a drone delivery system: a case study of Amazon prime air in the city of San Francisco. *Int J Adv Manuf Technol* **95,** 3141–3153. https://doi.org/10.1007/s00170-017-1363-1

Šipoš, D., & Gleich, D. (2020). A lightweight and low-power UAV-borne ground penetrating radar design for landmine detection. *Sensors*, *20*(8), 2234. https://www.mdpi.com/1424-8220/20/8/2234

Süzen, A., Duman, B., Şen, B. (2020). Benchmark analysis of jetson tx2, jetson nano and raspberry pi using deep-CNN. *International Congress on Human-Computer Interaction, Optimization and Robotic Applications (HORA), Ankara, Turkey.* pp. 1-5, https://www.researchgate.net/publication/342571472_Benchmark_Analysis_of_Jetson_TX2_Jetson_Nano_and_Raspberry_PI_using_Deep_CNN

Statista. (2020). Most used languages among software developers globally 2020 | Statista. Retrieved 22 October 2020, from https://www.statista.com/statistics/793628/worldwide-developer-survey-most-used-languages/

Xu, S., Savvaris, A., He, S., Shin, H., Tsourdos, A. (2018). Real-time implementation of YOLO+JPDA for small scale UAV multiple object tracking. *2018 International Conference on Unmanned Aircraft Systems (ICUAS), Dallas, TX*, pp. 1336-1341, https://www.semanticscholar.org/paper/Real-time-

Implementation-of-YOLO%2BJPDA-for-Small-UAV-Xu-Savvaris/84364190d6859067d0d26e4c8eb680627648c058

Ying, R. (2020). Airport and ship target detection on satellite images based on YOLO V3 network. *Lecture Notes in Electrical Engineering Proceedings of the 6th China High Resolution Earth Observation Conference (CHREOC 2019),* 167-174. https://link.springer.com/chapter/10.1007/978-981-15-3947-3_12

Yu, D., Xu, Q., Guo, H., Zhao, C., Lin, Y., Li, D. (2020). An efficient and lightweight convolutional neural network for remote sensing image scene classification. *Sensors (14248220)*, *20*(7), 1999. https://www.mdpi.com/1424-8220/20/7/1999

Zamri, N. M., Ling, G. F., Han, P. Y., Yin, O.S. (2019). Vision-based human action recognition on pre-trained AlexNet, 2019 *9th IEEE International Conference on Control System, Computing and Engineering (ICCSCE)*, 1-5, https://www.researchgate.net/publication/340687785_Vision-based_Human_Action_Recognition_on_Pre-trained_AlexNet

Zhang, F., Li, Q., Ren, Y., Xu, H., Song, Y., Liu, S. (2019). An expression recognition method on robots based on mobilenet v2-ssd. *6th International Conference on Systems and Informatics (ICSAI)*, pp. 118-122, https://www.researchgate.net/publication/339558970_An_Expression_Recognition_Method_on_Robots_Based_on_Mobilenet_V2-SSD