Trinity College Dublin
Coláiste na Tríonóide, Baile Átha Cliath
The University of Dublin

School of Computer Science and Statistics

Drones for active device inspection and sensing in complex contexts

Valerie Mai Brosnan

April 19, 2022

A dissertation submitted in partial fulfilment
of the requirements for the degree of
MAI (Computer Engineering)

Declaration

I hereby declare that this dissertation is entirely my own work and that it has not been submitted as an exercise for a degree at this or any other university.

I have read and I understand the plagiarism provisions in the General Regulations of the University Calendar for the current year, found at http://www.tcd.ie/calendar.

I have also completed the Online Tutorial on avoiding plagiarism 'Ready Steady Write', located at http://tcd-ie.libguides.com/plagiarism/ready-steady-write.

Signed: ___________________ Date: ___________________

Abstract

Wind energy is an effective low-cost and clean energy resource. To ensure further advancement and optimisation of wind turbines, there is a requirement for the development of structural health monitoring systems so that preventative and protective maintenance can be carried out. Typically access to wind turbine blades is very difficult and requires an industrial climber or a crane. This process can be very dangerous and increases the maintenance cost and risks associated with wind turbines. During the inspection process there is also a disruption in energy production as there is a requirement to shut down the wind turbine for inspection.

Drones have been used for inspection purposes and structural health monitoring tasks for many years now. This research looks at modelling the flight characteristics required to enable a drone to visually inspect the blades of a wind turbine and land on a moving surface. This will enable imagery and additional information to be collected from the blade of the turbine without requiring the turbine to be switched off. It will also remove the need for constant hovering by the drone which will optimise the battery life.

To achieve this, an autonomous system is developed which enables a drone to perform surveillance such as identification, recognition and tracking of a moving target. In this research, autonomous control is established over the Dagu robot, where the buggy can search, identify, and approach a target. The Dagu robot is used to simulate and model the control of a drone to increase the reliability of the system design. Computer vision methods are used to reliably track a moving target are investigated and tested to identify a robust, reliable algorithm. A Convolutional Neural Network is developed with a validation accuracy of 96.92% when tracking moving model cars in the horizontal plane.

This research has the potential to make major contribution to the expanding field of wind energy, by improving monitoring methods which will help support the early detection of faults, resulting in a reduction in the requirement for unscheduled maintenance, a reduction in wind turbine life-cycle costs and ultimately a reduction in the cost of energy.

Acknowledgements

I would like to express my gratitude to my supervisor Dr. Ciarán McGoldrick for taking the time to provide guidance, expertise , encouragement and support throughout this project. His advice has been invaluable, he never hesitated to answer any of my queries and I am very thankful for his patience and insight.

I would also like to thank Ronan Healy for his assistance in procurement of hardware components.

Finally, I would like to thank my friends and family for their help and encouragement over the last year.

Contents

List of Figures

List of Tables

1 Introduction

This project investigates the development of an autonomous system which enables a drone to perform surveillance such as identification, recognition and tracking of a moving target.

This system can optimise wind energy production by improving structural health monitoring methods so that preventative and protective measures can be taken to avoid failure of wind turbine blades.

1.1 Motivation and Background

There is significant motivation for this work. Wind energy is currently the largest contributing resource of renewable energy in Ireland. This area is predicted to grow rapidly as wind energy is the cheapest form of renewable energy and the cost of fossil fuels is rising. This research has the potential to advance this growth.

Wind turbine blades are highly susceptible to damage and often this is difficult to predict due to the location of wind turbines in harsh environments. Visual inspection is widely used but this method requires an industrial climber and trained personnel, making it very time consuming and expensive. These methods also require the wind turbine to halt causing a disruption in the production of energy.

With the focus on reducing energy conversion costs and meeting higher energy demands, there is an increasing need for periodic damage prognosis and condition-based monitoring of these blades. This research will provide a safe and low-cost method to monitor wind turbine performance, to ensure reliability and safety detecting the evolution of damage, and predicting performance deterioration. As the developed system is centred at landing on a moving target, this will also enable imagery and additional information to be collected from the blade of the turbine without requiring the turbine to be switched off, thus avoiding any disruption in energy. This method also removes the need for the constant hovering of the drone, optimising the battery life.

1.2 Objective

The primary objective of this project is to develop a robust control system to enable a drone to autonomously identify, recognise and track a moving target. Key considerations in this design are robustness, reliability and accuracy. The project aims to develop system control in a simplified environment to gain a deep understanding of the logic required before implementing the designed system onto a drone.

1.3 Approach

To develop this system the project is two main areas autonomous control and image processing.

Simulate drone control and develop an autonomous system in a simplified environment. Use this to transverse all potential scenarios in a single plane developing a control strategy to respond to them. A dagu robot was used to model the drone.

Develop a robust and reliable image processing algorithm to perform surveillance such as identification, recognition and tracking of a moving target.

Implement the autonomous system onto the drone and further test the control strategy.

1.4 Challenges

There were many challenges encountered in this project due to the broad range of topics covered.

A major challenge encountered in this work was the control strategy design could not be tested on a drone as planned due to procurement issues. Unfortunately there were issues in finding a suitable battery for the drone ordered. A battery could not be found which would arrive within the timeline of this work. As a result the final testing of the control strategy could not be applied to the drone. To overcome this challenge each aspect of the control strategy was tested independently.

Although the buggy was selected as I had previously worked with the hardware, many challenges were still encountered with the hardware. Initially the motors pins were not connected to the H-bridge, a lot of time was spent to correctly identify and connect the pins. The primary challenge when working with the buggy was that the motors responded differently when provided with the same power. The left motor was significantly more powerful than the right motor, to correct for this a corrective factor was introduced to prevent the buggy constantly turning, however this regularly needed to be modified as the motors would perform differently as the battery decrease. Minor challenges with hardware which were time consuming to resolve were also faced. These included setting up the XBee for wireless communication and working with the Pixy camera.

Another challenge encountered in this work was accessing aerial imagery of wind turbine farms to train a classifier. This lead to the creation of novel, experiment-specific datasets.

1.5 Report Outline

The remainder of this dissertation is structured as follows:

- **Chapter 2: State of The Art**
  This chapter provides and introduction into the topics relevant for this work. It starts with an introduction into wind energy and wind turbines. This is followed by an overview of current methods used for Structural Health Monitoring of wind turbines. Unmanned Aerial Vehicles are then introduced and methods to achieve autonomous flight are analysed. Finally, the security considerations associated with autonomous flight are then considered along with the ethical dimensions of this work.

- **Chapter 3: Design**
  This chapter outlines the design considerations for each component of the project and explains why each was important.

- **Chapter 4: Implementation**
  This chapter is an in depth explanation of how each component of this work was implemented to meet the design specifications.

- **Chapter 5: Evaluation**
  This project evaluates the implementation of the project by testing the reliability, accuracy and robustness of the implemented design. These are evaluated through experiments to test each implementation.

- **Chapter 6: Conclusion**
  This chapter is a summary of the main findings and contributions of this work. The future work is then outlined based on the limitations of this work. Before concluding there is a brief reflection from lessons learned through this project.

2 State of the Art

The aim of this chapter is to position the work of this project in the domain by providing a comprehensive analysis of the relevant research done in the field and providing a critical evaluation, and to identify avenues of future exploration which this project can address. This begins by looking at wind energy and structural health monitoring of wind turbine blades in order to emphasize the value of this research work, and highlight its positive impact on wind energy.

The computer vision issues are investigated, and methods explored for visual inspection by drones will be outlined. The final area explored in this topics is methods to fly and control a drone. This research provided an essential understanding to complete the project.

2.1 Wind Energy

This section provides a brief introduction to wind energy as a renewable, clean energy resource and emphasising why it is critical to be able to provide a low cost, efficient monitoring system to maintain a wind turbines energy production and provide health monitoring to enable preventative maintenance to be carried out.

There has been significant research, development, investment and deployment in renewable energy, as it is an effective low-cost and clean energy source. Wind turbines occupy a prominent place in the clean energy landscape, and it is accepted as the leading contributor to Ireland's green energy supply with strong future growth projections.

Due to the country's geographical position at the edge of the Atlantic, Ireland is perfectly located to harness the strength of the wind and to reduce our energy dependence on the rapidly depleting and environmentally damaging fossil fuels. Wind energy provides an emissions-free, clean and renewable energy source. It is currently the largest contributing resource of renewable energy in Ireland. In 2020 Wind Energy provided 38% (4) of Ireland's electricity, contributed to 86% of Ireland's renewable electricity(5), and thus wind energy contributed 36% to Ireland's total energy demand(5). This area is also predicted to grow rapidly as wind power is the cheapest form of renewable energy, and the cost of fossil fuels is rapidly increasing. To support this growth, it is essential to improve the structural health monitoring systems of wind turbines.

2.2 What is a Wind Turbine?

A wind turbines is a typical mechatronics system. Wind turbines operate on the principle that the energy in the wind turns three blades around a rotor. As the blades turn the rotor, they spin a shaft that connects to the generator, which consists of a conductor surrounded by magnets. The magnets spin around the conductor and convert the kinetic energy from the wind into electricity.

Wind turbines can be classified as Vertical Axis Wind Turbine(6) (VAWTs) or Horizontal Axis Wind Turbines(6)(HAWTs) based on their rotational axis. In a VAWT, the rotational axis is vertical to the ground, while the axis of a HAWTs is horizontal or parallel to the ground. The standardized design is typically a horizontal axis wind turbine (HAWTs) with one, two or three blades.

A HAWT is a wind turbine in which the main rotor shaft is pointed in the direction of the wind to extract the power. Each turbine consists of a tower, a nacelle and the blades. The tower is typically made of steel, and raises the blades into the strongest airflow. Most HAWTs are comprised of three blades, as this provides the most energy conversion while limiting noise and vibration. The gearbox, high speed shaft and generator are all housed in the nacelle. Energy is harnessed by the rotor. The rotor receives the energy from the wind which produces a torque on the low-speed shaft. This low-speed shaft in turn transfers the energy to the gearbox, which then converts the slow speed of the spinning blades into a higher-speed rotary motion which turns the drive shaft quickly enough to power the generator. The structure of a wind turbine is illustrated in Figure 2.1

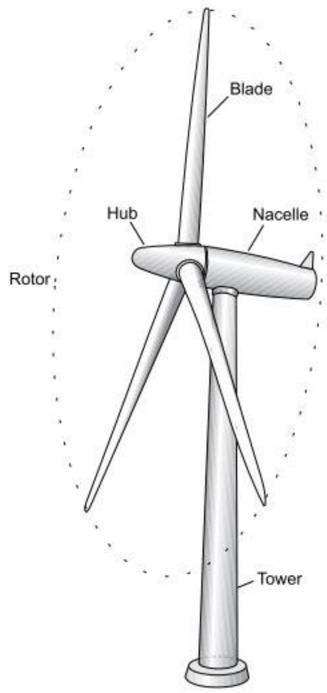


Figure 2.1: Basic features of a wind turbine(1)

Wind turbine blades are made of fiberglass, a reinforced plastic material that is embedded with glass fibres which are randomly laid across each other and held together with a binding substance. Fibreglass allows high strength at a low weight, so that longer and more efficient rotor blades for wind turbines can be manufactured in a cost effective way(7).

But fiberglass turbine blades can be damaged by moisture absorption, fatigue, wind gusts or lightning strikes(8). Often these are difficult to predict as aerodynamic interaction between different turbines can cause unpredictable and excessive loads on the blades, which can accelerate the predicted fatigue damage to the blade(8). The failure of one blade may damage nearby blades and wind turbines, increasing the total cost of the damage(9).Defective blades are rarely replaced due to the high cost of manufacturing, thus it is vital to prevent failure which can be achieved by continuous monitoring(9).

Wind turbine farms can be categorized as land-based wind conversion systems and offshore wind conversion systems. This classification is based on where the wind farm is located. In the case of offshore wind farms, the turbines are placed over open water where high-speed winds are available. This wind is then used to generate power. Land based wind farms are located on land typically in exposed windy areas. This research work looks at land-based farms, because the price of electricity produced on land-based plants is significantly less expensive than that from offshore farms, due to higher costs of installing and maintaining the turbines in offshore location (5).

Since the early 2000's wind turbines have gradually grown with both the height of the tower and the blade length increasing(10). This is resulting in taller turbine towers to capture more energy, as wind speeds tend to increase as altitude increases. Turbines with larger blades enable the turbine to sweep a larger area, thus capturing more wind and producing more power. A turbine's rotor diameter is the diameter of the circle swept by rotating the blades. As the rotor diameter increases, the capacity or maximum power rating of a wind turbine also increases, demonstrating a positive correlation between the rotor diameter and capacity as illustrated in Figure 2.2.

However, there are constraints restricting the size of a wind turbine(3). The vibration of blades and the tower is a limiting factor when designing larger wind turbines. The increasing size of towers and blades has led to vibration issues due to the dynamic nature of the environment in which the structures operate and the choice of materials for the turbines. Due to large size of the towers and the high velocity of the wind speed, significant vibration occurs in the towers and the blades. The HAWT towers experience further vibration problems due to the heavy load of the nacelle at the top of the tower. This vibration in the system causes a reduction in the efficiency and must be monitored carefully to optimise the performance of the wind turbine.

Vibrations are characterized as in-plane or out-of-plane vibration as wind turbines vibrate on both axis(3). This is illustrated in Figure 2.3. In-plane vibration is also known as edge-wise vibration, refers to vibrations that occur in the plane of rotation of the blades, while out-of-plane vibrations also known as flap-wise vibration, refers to the vibrations that occur outside of the blade rotation plane.

There are many methods being explored to mitigate the issues that arise due to these vibrations. One such method is the use of Active Tuned Mass Dampers (ATMD). ATMDs are damping units placed in structures where a mass is actuated to move out of phase with respect to the movement of the structure to reduce vibrations(11). However edgewise vibrations are

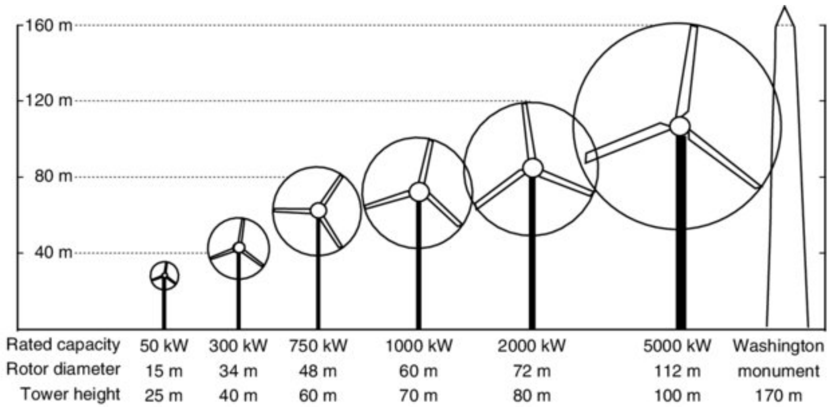


Figure 2.2: The capacity relationship with with rotor diameter and tower height(2)

more difficult to detect (12). This emphasises the need for the development of an efficient Structural Health Monitoring component of these turbines.

Particularly with the increase in the size of wind turbines, maintenance becomes more challenging, and it is becoming more important to improve structural health monitoring of the systems to optimise the performance of wind turbines.

2.3 Structural Health Monitoring

To ensure future growth in the industry, wind industry technology must continue to evolve. A key component to the further advancement and optimisation of wind turbines is the development of structural health monitoring systems (13), as this plays a vital role in achieving reliable, safe and economic operation of wind turbines. Wind turbine manufacturers, owners and operators may all benefit financially from SHM technology which can provide an indication of the reliability of each wind turbine throughout its life cycle.

This research is centred around the SHM of the blades of the wind turbine. The blades are the most critical components in the wind turbine system as they are essentially the collectors of wind energy. They are also the components that are most susceptible to damage. With the focus on reducing energy conversion costs and meeting higher energy demands (with the use of larger turbine blades), there is an increasing need for periodic damage prognosis and condition-based monitoring of these blades(14).

This section will provide a brief introduction into techniques currently used for the structural health monitoring of wind turbines.

The term 'health' in relation to a wind turbine encompasses the loading, damage and operational capability (i.e. life at the rated performance) of a turbine(15). Maintenance of wind turbines can be corrective or preventative. Scheduled maintenance is typically carried out on a turbine twice per year requiring approximately 24 hours per turbine(15). However unscheduled maintenance is approximately more costly requiring on average 130hours per turbine(16). The implications of this research work can improve monitoring methods which will help support early detection of faults and as a result reduce the need for unscheduled maintenance, reducing the wind turbine life-cycle costs and ultimately the cost of energy.

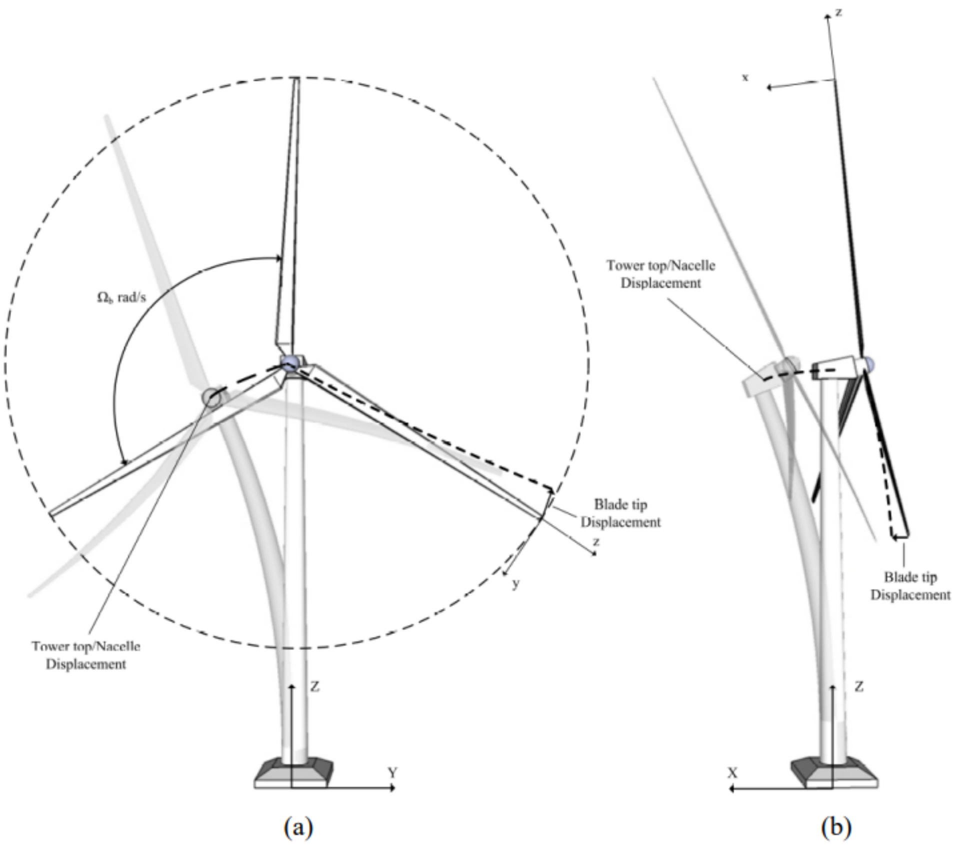


Figure 2.3: Wind turbine vibration,(a)in-plane (b)out-of-plane(3)

The aim of a structural health monitoring system is to detect minor damage sites before they can combine and propagate, which would eventually result in failure of the blade. Health monitoring of the rotor blades and timely identification of potential failure areas can prevent failure of the entire HAWT. This research work will be centred on Non-Destructive Testing (NDT) methods as they leave the blade under assessment undamaged.

2.3.1 Visual Inspection

Visual inspection is widely used as part of routine maintenance of wind turbine blades. Based on the scale of the blades the inspection process is time consuming and the accuracy of the results is highly dependent on the inspector's skill. For effective visual inspection to be performed, there are many considerations. Visual Inspection is only possible at close range, using either an automated tower climbing machine or by suspending from the top of the nacelle with ropes. With this method workers safety is a concern as the 'sky workers' perform the inspection while suspended by a rope attached to the turbine or supported by a platform(17). Alternatively visual inspection can be carried out from the ground level to identify visible damage using binoculars or digital cameras.

Visual inspection techniques have limitations as they can not identify inner damages of the blade. Another issue associated with this method is that it is difficult to standardise results that are recorded by inspectors. Subjective analysis and human error can give rise to results that may not be very reliable. Traditionally this inspection method is very expensive in terms of both man hours and structure down-time. The development of a suitable automated and reliable monitoring system that is efficient and accurate is very desirable.

This project is centred around monitoring the structural health of the moving blades of a wind turbine. As the diameter of these blades increases and the number of wind farms in Ireland grow, there is a growing need for SHM. It is vital to provide a safe and low-cost method to monitor wind turbine performance to ensure reliability and safety, detecting the evolution of damage, and predicting performance deterioration.

Conventionally Non-Destructive Testing (NDT) techniques require close proximity between the sensor and the blade (18). Access to a blade by industrial climber or a crane is very difficult and expensive. This process can also be very dangerous and increases the maintenance cost associated with wind turbines. This emphasises a gap in the industry for the need of a safer and cheaper approach which can be achieved with Unmanned Aerial Vehicles (UAVs).

Research(8) has shown that predicting the exact fatigue life of a blade is very difficult, and it is also difficult to tell the extent of fatigue damage that might have occurred to a blade. In this experiment faults were only detected on the path between the sensor and the actuator on a WT while it was switched off. This project attempts to advance this process by enabling a drone to land on the moving turbine blade, so that information can be collected without the wind turbine being disabled, which would cause a disruption in the production of energy.

Other NDT testing methods that have been investigated to monitor the blades of a wind turbine are:

- UT- Ultrasonic Testing
- AE-Acoustic Emission
- FBG- Fiber Bragg Grating strain sensors
- Tap tests
- Ground based Radar (GBR)

A fast, accurate and cost-effective method of SHM is Acoustic Emission and Acoustic Ultrasonics. This enables detection of faults long before the structural integrity is compromised, and structural failures occur without applying further strain to the blade. However, for these methods to be successfully implemented close proximity to the blade is necessary.

2.3.2 Ultrasonics

Ultrasonic applications have shown great potential for wind turbine blade inspection. There are many different techniques which use ultrasonics to improve structural health monitoring.

Ultrasonic Echo

One such ultrasonic echo technique works by transmitting short-duration ultrasound pulses into the region that is being explored. The echo signal resulting from scattering and reflection is detected and recorded. The depth of a reflective structure can be inferred from the delay between pulse transmission and echo reception,thus enabling the geometry of defects to be detected and their approximate dimensions can be estimated. In order to carry out this technique, the pulse must be transmitted into the region being explored which could be a potential application of this project(13).

Lamb Waves

Further research(19), identifies an ultrasonic detection technique that is based on Lamb waves that can be applied for SHM. Lamb waves are elastic waves whose particle motion lies in the plane that contains the direction of wave propagation and the normal plane.

Lamb waves make it possible to investigate large areas of a structures in a quick and reliable way. This is because Lamb waves have many unique properties such as a high sensitivity to the properties of materials, sensitivity to cracks at different depths and the ability to propagate over the entire thickness of an object(19).

This research(19) identifies and characterises faults using lamb waves and specific signal analysis algorithms. However it does not identify a 'superior' algorithm process the results. The signal processing method contributes to the limitations of this method as it is very complex and subject to external noise and further research is required to optimise the SHM using Lamb waves(19). What is again evident with this research (19) is it is *'theoretically'* very promising but further research is required in order to introduce apply this to the SHM of WT.

Laser Ultrasonic Imaging

A laser ultrasonic imaging technique was proposed specifically for rotating blades(20). This method does not require the WT to be switched off. This is implemented by sequentially generating ultrasonic waves at multiple points by using a scanning excitation laser beam. A surface mounted piezoelectric transducer at a single point is used to measure corresponding responses. In addition this method implements an advanced signal processing technique for automated visualization of a subsurface defect. The results demonstrated that this ultrasonic imaging system can be successfully constructed even with the fast-rotating speed and complex geometry of the blade. Damage that was invisible from the scanned surface was successfully identified, and its visibility has been enhanced using standing wave filters. This research(20) results were found to be promising. However the experiments were conducted in a lab on a rotating metal fan. Further inspection is required to determine if these results could be achieved on site.

2.3.3 Vibration response

There are a variety of Non-Destructive Testing (NDT) methods that look at the vibration response of wind turbine. Ghosal (8) introduces four methods for detecting damage on wind turbine blades. All of these methods are based on measuring the vibration response of the blade when it is excited using piezoceramic actuator patches bonded to the blade. These vibration measurements are useful as they can identify damage inside the blade without having to map over the surface of the entire blade with a sensor.
The four methods investigated are:

1. Transmittance Functions - TF
2. Operational Deflection Shapes - ODS
3. Resonant Comparison - RC
4. Wave Propagation -WP

To test these damage detection methods, an experiment was carried out on an 8-foot-long section of a fiberglass wind turbine blade which was supported by a rope and elastic cord. The results concluded that the Resonant Comparison method was the only one of the four methods suitable for damage detection during operation which is the goal of this research work. This method involves exciting the blade to its resonant frequency and measuring the response of four sensor patches. The results are recorded on an oscilloscope. To detect damage, the results are compared to 'healthy data', the results on a structurally sound blade.

The results reveal this method to be practical as it can be used for damage detection with minimal historical data, and it can be used on an operating wind turbine.

The key issue identified with this method is that further testing is required for different damage types and on a larger turbine blade. Large modern wind turbines in Ireland have rotor diameters ranging up to 130 metres (4), so further testing must also be carried out on blades of a larger size to ensure this method can be scaled to size. Ghosal's paper does not address the lifetime of the piezoceramic actuator patches. This would need to be investigated further as uninterrupted performance of the patches is a requirement of a reliable monitoring system.

2.3.4 Visual Imaging

Current methods for structural health monitoring of wind turbine are time consuming, and the operation time of the turbine is reduced as the turbine is required to halt while inspecting. These problems are overcome by the inspection through drones. Image analytics can be utilized to develop a solution that employs deep learning Convolutional Neural Networks (CNN) to train on lots of images captured by drone such that after the training, the model can be used to classify the new input image of WTB and classify them as damaged or not damaged.

This research work looks at using an Unmanned Aerial Vehicle to navigate to the turbine to collect images of the blades of the wind turbine and land the drone on the surface of the blade so that further analysis can be carried out. This method of monitoring can reduce the maintenance and inspection time and provides less risk for the inspection of the WTB for both the structure and maintenance workers.

Researchers(21) have successfully implemented a wind turbine blade damage classification and detection system for different classes of damage by training a model from an image dataset prepared by image augmentation methods and manual editing. This method acquires images of the blades of wind turbines, and damages are then classified so that the particular area of damage can be located (21). The method produced accuracy of 99.4% for binary classification and 91% accuracy for multiple class fault classification(21).

Using drones to capture images of wind turbine in operation will reduce maintenance costs and increase productivity. Unmanned Aerial Vehicles can be used to improve the structural health monitoring in a non-intrusive approach. UAVs provide a safe and more efficient alternative to visual inspection methods as they can provide access to areas which may be deemed 'high risk' due to hostile working conditions or as they cannot be safely accessed. UAVs can also be used in conjunction with acoustic emission, ultrasonic and visual imaging to further advance the capability of SHM. This project intends to look at methods to successfully landing a drone on the WTB to allow for further analysis to be carried out.

Drones have previously been used for non-destructive evaluation of wind turbine structures. The potential applications for unmanned aircraft in NDT are largely focused on defects detection, damage analysis and condition monitoring(22).

Drones with attached sensors have an existing role in the SHM of civil infrastructure.There has been extensive research into an integrated framework approach. (23, 24). The research work of Kada(25) in 2011 emphasises several key design techniques which are vital to flight control development, stressing that it is vital to consider flight control in the context of dynamic modelling, control and model analysis, simulation, control design and real-time implementation.

Visual inspection has traditionally been used for structural health monitoring of civil infrastructure However, conducting this visual inspection by trained personnel increases the cost of upkeep and maintenance of the infrastructure. UAVs are the safest and most efficient alternative to traditional inspection methods as they provide access to areas that may be deemed as 'high-risk' due to hostile work condition and areas that cannot be safely accessed. Thus, drones equipped with high-resolution cameras have a growing application for performing inspection and surveillance of wind turbines.

As the drone is equipped with a high-resolution camera, it has the capability to employ computer vision techniques in order to track and detect specified targets and to record high-resolution imagery of the structure which can then be analysed to detect faults. UAVs are cost efficient due to the broad range of low-cost commercial autopilots available removing the need for synthesizing, implementing and validating a flight control system as is required for crewed aircrafts.

Research(22) investigates the use of UAV for remote building inspection and monitoring; the infrared building inspection with UAV. Research(26) has been carried out where UAV-based laser scanner and a multispectral camera data can be used in building inspection.

UAVs can be used to inspect structures using visual imaging and they also have the capability to incorporate the current sensory SHM methods and enchance them making inspections faster, cheaper and safer.

2.4 Unmanned Aerial Vehicle

Unmanned area vehicles, more commonly known as drones, have a wide variety of commercial applications. They can be flown remotely using a controller, or autonomously. There is a wide range of commercial applications of UAVs, including logistics, military operations, public security, traffic surveillance and monitoring. The use of commercial drones is becoming more widespread, especially for aerial photography, surveying and inspections as well as safety and security monitoring activities. There is a growing demand for reliable and low-cost UAV systems due to the broad range of applications. Images captured by drones flying in areas which are difficult to access otherwise will fill a gap between expensive, weather-dependent and low-resolution images provided by satellites, and car dashcam images which are limited to human-level perspectives and the availability of accessible roads(27).

Visual inspection has traditionally been used for structural health monitoring of civil infrastructure However, conducting this visual inspection by trained personnel increases the cost of upkeep and maintenance of the infrastructure. UAVs are the safest and most efficient alternative to traditional inspection methods as they provide access to areas that may be deemed as 'high-risk' due to hostile work condition and areas that cannot be safely accessed. Thus, drones equipped with high-resolution cameras have a growing application for performing inspection and surveillance of wind turbines. As the drone is equipped with a high-resolution camera, it has the capability to employ computer vision techniques in order to track and detect specified targets and to record high-resolution imagery of the structure which can then be analysed to detect faults. UAVs are cost efficient due to the broad range of low-cost commercial autopilots available removing the need for synthesizing, implementing and validating a flight control system as is required for crewed aircrafts.

2.4.1 UAV Architectures

A quadcopter is a form of UAV with four symmetrical rotor propellers. Figure 2.4 shows a quadcopters schematic.

(a) '+' Configuration of Drone (b) 'X' Configuration of drone

Figure 2.4: Drone configurations

Quadrocopters are especially advantageous because of their small size, low cost and low control complexity(28). A quadcopter can typically have an 'X' configuration or a '+' configuration as seen in Figure 2.4. The 'X' configuration has two front motors and two back motors as seen in Figure 2.4b, while the '+' configuration has a single front, back, left and right motor as illustrated in Figure 2.4a. The 'X' configuration is more widely used as most quadcopter builds have a front facing camera which would be impeded by the front motor if a '+' configuration was selected. Thus, for this research work a 'X' configuration UAV was selected.

Each rotor propeller creates an upward thrust which is highly manoeuvrable by controlling the speed variation which enables the quadcopter to take-off, land and hover. A quadcopter is a Vertical Take Off and Landing (VTOL) aircraft, having four vertically orientated propellers that can be tilted for movement while in flight. The drone is primarily controlled by throttle, pitch, roll and yaw as shown in Figure 2.5.

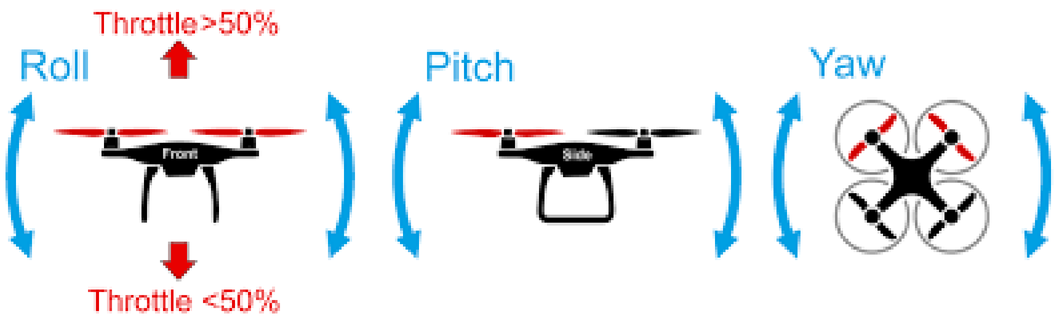


Figure 2.5: Input effects on the drone

- **Throttle** determines the overall speed of motors. This directly relates to the height at which the drone flies. At 100% throttle the motors will spin at maximum speed and the the drone will travel at maximum height.
- **Pitch** determines the forward/backward angle that the drone makes with the ground. With a high pitch value, the drone will angle itself forward, by increasing the speed of the back motors and begin to move in that direction as the lift vector of the motors has also been angled in the forward direction.
- **Roll** determines the left/right movement of the drone. This is achieved by increasing (or decreasing) the speed of the left propeller and by decreasing (or increasing) the speed of the right one to cause a torque in the x-axis.
- **Yaw** is similar to roll. It is rotation about the z-axis which corresponds to clockwise/counter-clockwise rotational movement of the drone. This is achieved by changing the propellers speeds to unbalance the overall torque causing the quadcopter to rotate in the z-axis.

Researcher has found(25) outline the need for the improvement in the modelling, testing and flight control of UAVs so as to increase their reliability and performance during autonomous flight, proposing aspects which this research work will investigate. Methods of shortening the development cycle of UAVs to improve system reliability and robustness of the flight control system have been investigated(25) and why it is important to develop an integrated framework for the flight control design process. The (25) process consists of a set of design tools that enable control engineers to rapidly synthesize, implement, analyse and validate a controller design. This concept is one which this research work intends to implement. There had been much research into an integrated framework approach. (23, 24). The research provided many interesting insights into providing a systematic approach for the different processes in model-based flight control development.

2.5 Target Identification and Tracking

In order to develop a highly robust control strategy, a vital component is the identification and tracking of the target. This section looks at methods previously used for target tracking and recognition systems. Target tracking and recognition has been a well-studied topic(29) and there have been significant advancements recently using deep neural networks(30).

The development of computer vision and machine learning techniques enable low-cost UAVs to perform complicated learning tasks(28). For a UAV to carry out image recognition, high quality images are required. In addition to common tracking and recognition issues, there are additional technical challenges such as background, motion blur, and low-resolution images which must be resolved for UAV based tracking and recognition. Research reveals(28) a complete framework of UAV based tracking and identification has not been fully developed yet. The additional challenges after basic recognition and tracking which must be resolved are:

1. To develop *real-time* algorithms for varying background video processing, as targets will be recorded within a moving background.
2. The target's movement could be very fast compared with the speed of the UAV camera responses. In addition, camera vibrations caused by aerial turbulence may cause blurred images and missing targets.
3. The image resolution of targets is often very low due to long distances between the UAV and the targets. The detection and identification of targets with a high accuracy using low resolution imagery is a key challenge.
4. Target occlusion with a changing background must also be considered as it can be difficult to determine if the target leaves the Field of View (FoV) or it is just occluded by other objects.

The erratic motion of targets together with their changeable outline means that they are conventionally very difficult to model. This process can be simplified by detecting the colour of the target and looking for this in a later frame; identification and tracking can thus be achieved. A target can be identified by defining the overall image content in terms of RGB derived opponency or the HSV colour space. This allows the identification of a colour space which can contribute to the identification of salient regions. The extraction of interesting information(identifying the target) from an incoming sequence of images is visual saliency, this simulates the human visual system to perceive the scene. The computation requires an object detection process which extracts the salient region from the image(28).

The Red, Green, Blue (RGB) color format represents a colour by additive mixing of the primary colours. The HSV colour space represents colours in terms of the hue which is the color portion of the model, Saturation (S) which represents the quantity of grey and their brightness Value (V). Using the HSV colour space gives more perceivable information for extracting interesting regions than the RGB derived opponency colour(31). This process separates the luminance component from the colour information which is helpful for the various object detection applications, and so is more robust to lighting changes than the RGB color space(31).

Tracking and identifying an object by colour is a simple method to track a target's motion in real-time. However it does have many limitations such as distinguishing between different targets.

No single algorithm can achieve the best performance in terms of all the tracking and recognition metrics(32), but a trained model may provide more reliable functionality. For example, Convolutional Neural Network based methods have been found to yield better recognition performances but they have a high computational cost.

To develop a robust system, highly distinctive feature representation at each frame and accurate feature registration among frames is required(28). A successful feature descriptor should have the capability of scale-invariance, rotation invariance, robustness against noise, a sufficient representation of blur images, and a high computing speed.

A ConvNet model (ST- ResNet) including appearance and motion stream models can be utilized to recognize the actions of targets(33). Deep convolution neural networks are employed to achieve target detection and recognition. You Only Look Once (YOLO)(34) is a object detection system which employs a deep learning model and achieves good performance in real-time. It is a simplified network which can be used to obtain bounding boxes in the images captured by the UAV. A dataset is used to train the YOLO model.

This is a supervised machine learning network in which we are learning to predict output/target values from a labelled dataset. In order to implement this, a labelled data is required where each image is labelled with an associated tag/number to identify certain classifications or contained objects. Labelled data is required for supervised learning.

The gathering and storage of large labelled datasets has been essential for the growth of growth of machine learnings, and the quality of the training data is vital to a models performance. Training a model requires a large labelled datasets. Many of the open datasets like Microsoft COCO and ImageNet have a low percentage of aerial images in the dataset, which is too small to represent the imagery and achieve a good performance. The dataset used to train the model should be a true representation of the data and so it should include aerial imagery, low resolution and motion blurred images. To improve the efficiency and effectiveness of the detection and recognition models(32) the model may be trained on an annotated video specific for the use case. However further consideration must be given to ensure the dataset is not too small as this may result in overfitting to the test data which would cause the classification software to perform badly as it has not been trained on a sufficient amount of data and cannot generalise well. Dataset selection is vital if using a machine learning model as if the dataset is unrepresentative, too 'noisy' and thus unreliable or if the dataset fails to capture the useful relationship the resulting model will perform poorly.

Using computer vision techniques, a reliable system can be designed to recognise and track a target, however careful consideration needs taken to ensure that it successfully recognises and tracks the target considering all technical obstacles.

2.5.1 Autonomous flight

Autonomous drones operate by using artificial intelligence powered navigation and operational software thus removing the need for a pilot. Robot autonomy is defined as the ability to perform intended tasks based on the current state and sensing, without human intervention(27).

There are three levels of increasing autonomy (27)

1. **Sensor-Motor Autonomy**
   This relates to the ability to translate high-level commands such as reach a given altitude, perform circular trajectory or follow GPS coordinates into low-level platform-specific combination of yaw, roll and pitch.

2. **Reactive Autonomy**
   Reactive autonomy is dependent on sensory-motor autonomy. This is the ability to maintain current position or trajectory in the presence of external perturbations. Examples of this include wind, maintaining a safe or predefined distance from the ground, coordinate with moving objects and take-off and landing. For this research work the robustness and reliability of the design is vital. Obstacle avoidance is emphasized as one of the most important components to provide reactive autonomy to the system.

3. **Cognitive Autonomy**
   Cognitive autonomy is also dependent on reactive autonomy. Cognitive autonomy describes a UAVs ability to perform simultaneous localization and mapping, resolve conflicting information, plan for system events such as battery recharge, recognise objects or persons and as the system is an AI system, learn from events.

This research work (27) identifies the considerations required when developing a autonomous system for a drone providing an essential understanding for designing the system this research work addresses.

The concept of this research work is to identify and track a moving target is similar to previous research (35) which provides an autonomous flight system for a drone using marker recognition. The proposed system (35) maintains a distance between the drone and the marker in flight. The system then estimates the distance from the marker and the drone by calculating the area of the recognized marker. While Kim (35) proposes this idea for autonomous control in replacement of a GPS, the proposed method may also be used in conjunction with GPS to increase the system reliability and provides interesting ideas for target identification which may be introduced in this research work.

Autonomous UAV's have been used for military surveillance(36) where the objective is to explore a field are and find and localize target positions, which is similar to the aim of this research work. The approach used by Ma'sum (36) builds an AI system. The system consists of three main modules.

The first module involves estimation of the robot position which begins with navigation data acquisition.

1. Navigation data consists of inertial measurement unit (IMU) which is represent of UAV orientation, and UAV velocity data.The data is processed by Extended Kalman Filter

(EKF) algorithm to determine the drone's position in real world space.

2. The second module is estimation of target position, which uses image data from the UAV camera.
3. Then, image data is processed using Adaboost classifier to detect target object appearance. If the target object is detected in the image frame, then coordinates of the object are saved for next process
4. Object coordinate information in the image frame, camera focus length data, and UAV altitude are used to determine target position relative to the UAV using the Pinhole algorithm. This provides the target position data in real world space.
5. Then, using the UAV's position data and the target position relative to the UAV data, the position can be determined. The robot position and target position can be visualized

This method was implemented in a simulation and results indicated that the performance of the object detection is 71% successful

While the research work(36) outlined is experimental, it provided interesting concepts to be considered when developing the research work of this system. The proposed future work(36) includes implementing an object recognition algorithm to enable the UAV to differentiate between two or more target objects. While the research work (36) work is primarily centred around surveillance in military applications, it is not examining target distinction. For this research project, target distinction is vital as the UAV must know which target has been identified to know which target it will be landing on. The work outlined(36) outlines a successful method for target identification and the further work can then be undertaken in this project to distinguish targets.

The research work outlined provides an essential basis for the requirements of constructing an autonomous Unmanned Aerial System (UAS). With autonomous control, the reliability of the system must be considered specifically for the case of using the drone imagery for flight control and navigation.

2.5.2 Security Considerations

This section outlines the primary security and privacy consideration associate with using Artificial Intelligence guided UAVs and the potential impacts on flight control.

The primary security and privacy consideration with UAVs is their vulnerability to adversarial attacks. Adversarial attacks are classified as evasion attacks. An evasion attack occurs at test time, where a clean target instance is modified to avoid detection by a classifier or cause misclassification.

This type of Adversarial attacks can be classified into two categories: targeted attacks and non-targeted attacks, both which would result in the target not being identified or being misidentified. Both are potentially highly dangerous, as they directly impact the results of a classification system.

A targeted attack is when the adversarial example is misclassified into a predetermined class y' that is different from the original class $y' \neq y$. For a non-targeted attack, the attack goal is to craft adversarial examples that will be misclassified, but the misclassification class is not required to be of a particular type. Thus, a non-targeted attack is considered successful if the input is predicted with any wrong label, while a targeted attack is only considered successful if the adversarial example is classified as the target class. Both targeted and non-targeted attacks are specifically tailored for the the specific network as they require the addition of an imperceptible noise to the sample.

A targeted attack would have a greater impact on our system, and the attacker could get the system to identify a different object as the target. For example, it could cause the CNN model to classify a plane as its target instead of the wind turbine blade. This type of attack poses a greater threat as this malicious attack could result in:

- Damage to both the misclassified target and the drone
- The potential to cause a crash
- Failure of target identification and tracking, rendering the developed system unsuccessful.

A non-targeted attack would also result in system failure, however its impacts tend to be less severe.

Adversarial attacks can occur by poisoning the training data and are known as poisoning based attacks. The attacker or malicious user injects false training data into the system, with the aim being to control the behaviour of the developed classifier; in this case an attacker may target a specific test instance. The implications of a poisoning attack will obviously effect the reliability of the developed system. Fortunately, these attacks can be identified during testing as they degrade the test accuracy.

Another form of poisoning attack that is more dangerous for developed classifiers is a targeted backdoor attack (37) which can cause the classifier to fail for special test cases. These type of attacks are more difficult to detect and thus defend against, and thus could potentially have catastrophic consequences. A backdoor attack maintains normal or desired performance on benign samples.Its misprediction is only activated by samples which are attached with triggers.

The strongest threat model identified for the developed system is a white-box attack, where the attacker has complete knowledge of the target model, including its architecture and all weights. A black-box attack is less dangerous as the attacker has no knowledge about the model and data used behind the services, and thus the attacker's capability is limited to allowed functions.

A white-box attack can have larger implications as in this case the attacker can examine the inner workings of the model and so is better positioned to identify vulnerabilities in the system. Thus far consideration has been given to attacks which alter the drone's perception of an environment. However it is also possible to alter the environment to confuse the deep learning system into misclassifying targets. This attack could result in the drone identifying and attempting to land on a target which it perceives as the target.

Adversarial patches can be printed out, added to any scene, photographed, and presented to image classifiers. These must be considered even when the patches are small. They can be placed anywhere within the field of view of the classifier and could cause the classifier to output a targeted class (38) or to misidentify the target, which could cause the drone control system to misbehave in an unexpected and potentially dangerous way.

This type of attack can be performed for both targeted and non-targeted attacks. As expected, the patch size required to reliably dupe the model in the black-box attack is significantly larger than those required to perform a non-targeted attack in a white-box setting.

Because an autonomous system operates without human intervention, it is susceptible to adversarial attacks which could go unnoticed. Adversarial attacks could result in severe impacts on a UAS, and this illustrates that while a autonomous system is desired, a distributed control system is essential for reliable UAV control. A distributed control system ensures that a user can override the autonomous control to prevent unexpected incidents, and in the case where one arises, to minimise its impacts.

2.6 Ethical Dimension

When investigating a flying drone, there are three key ethical dimensions to consider. These are the health and safety components, the privacy considerations and the impacts on society of the project.

As the drone is essentially an unmanned aircraft system, operated by remote control or operating autonomously, careful consideration has to be given to the risk of accidents and collisions. If a drone malfunctions or has a mid-air collision, this has the potential to be dangerous and harmful. The safety of individuals, structural entities, birds and wildlife and the general environment must be considered when operating the drone.

Due to the nature of the drone using a camera to identify objects and navigate to a destination, this may be considered as an invasion of privacy as people may be recorded without their consent. The camera may also be used to collect data while drones can be deployed autonomously. The abuse of drones to photograph a person without their explicit knowledge and consent, or to stalk an individual is an invasion of the individual's right to anonymity and privacy.

Another ethical dimension to be considered in the course of this project is the potential displacement of workers who are currently carrying out roles that may be automated in the future. While this project seeks to optimise performance and minimise the employment costs associated with the structural health monitoring of moving objects, the ethical considerations of potentially eliminating jobs from people who previously carried out this role are significant. The successful implementation of this technology could lead to unemployment, underemployment or decrease in wages for workers who previously performed structural health monitoring roles.

2.7 Summary

As the size and prevalence of wind turbines increases to meet energy requirements, maintenance becomes more challenging, and it is becoming ever more important to improve structural health monitoring of the systems to support the growth, reliability, safety and efficiency of wind energy systems. To optimise SHM, a method which does not require a disruption in energy production is required. UAVs can be used to provide unobtrusive monitoring using visual imaging and servoing. UAVs can be used to capture images of wind turbines in operation or in conjunction with other NDT such as acoustic emission and ultrasonics monitoring.

To facilitate further advancements in wind energy, a robust and reliable system can be developed using visual imaging to track the blades of a moving wind turbine to enable the UAV to land on the moving structure so that further analysis can be carried out. This research emphasised the need for improvement in the modelling, testing and flight control of UAVs so as to increase their reliability and performance during autonomous flight.

3 Design

This chapter will look at the design considerations of this work. The aim of this work is to develop a highly robust autonomous control system which can identify and track a moving target. In order to design this system a three tiered design strategy was used, Initially the functionality was initially modelled on a 'DAGU' dg48:1 robot, the buggy. The buggy was used to simulate and understand the requirements of a drone and the necessary logic to implement this. The buggy represents a subset of the drone. The buggy has two motors, can move in a single plane and can travel in 260 degrees. The drone has four motors, can traverse 360 degrees and has an additional 3D component enabling it to move up and down.

The next component involves developing a highly robust control strategy through computer vision to track a moving target.

The final tier involved extending this work and deploying the developed control strategy onto a drone. This would involve getting the drone to identify and land on a stationary target before tracking and landing on a moving target.

3.1 The Buggy

The design goal of this component was to identify the key logic required for autonomous control and simulate this control design with real-time implementation in a simplified environment.

The 'Dagu' robot consists of the basic shell of a buggy which is comprised of motor assemblies, battery holders and sensors. The buggy includes an Arduino microcontroller, an XBee chip, a Pixy camera with a built-in machine learning chip for object detection and a front-facing ultrasonic range measuring module as show in Figure 3.1.

The sensor signals from the ultrasonic detectors are connected as inputs to the microcontroller. The motor control commands are then formulated based on the inputs, with the motors themselves being operated using a H-Bridge motor controller circuit. The system level block diagram in Figure 3.2 reflects how all connections and communications were established.

The buggy as illustrated in Figure 3.2 consist of two motors and can travel in 260 degrees where the drone will have four motors and can travel in 360 degrees in the plane, while also having a 3D component with the ability to move up and down, and to tilt, yaw and roll.

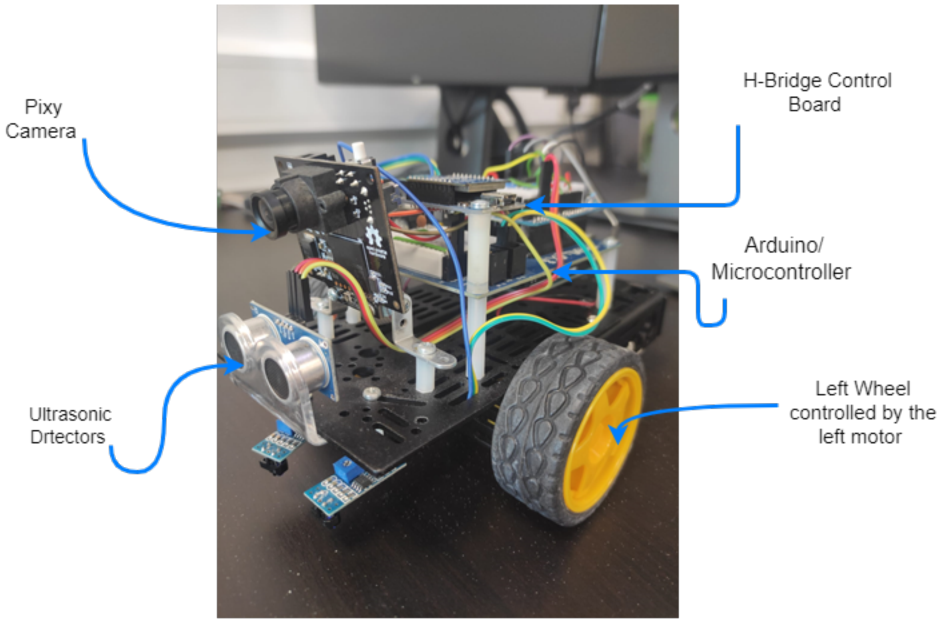


Figure 3.1: Components of Dagu robot Buggy

The design requirements specified for the buggy are:

- **Develop reliable control**
  This involves setting up the buggy and its motors and testing to insure the system acts as expected.
- **Search and identify a target**
  Developing an autonomous system to search and identify a target
- **Approach target safely**
  Once the target has been identified the autonomous system will approach the target, and stop prior to colliding with the target.
- **Dynamically adjust tracking**
  In the event that the target moves, the system responds accordingly and re-identifies the target
- **Provide communication between the buggy and the user interface:** The system sends back status updates to a user interface to provide updates on its progress, for example *'Searching for target'*, *'Target identified'*. Although it is an autonomous system, introducing a 'user' override is an essential safety feature, so the user can change the status of the buggy. This is achieved by ensuring the buggy responds appropriately to a control request from the user.
- **Identify limitations**
  Investigate the capabilities of the buggy such as the field of view of the camera and how the capabilities effect identification of the target.

The design strategy was to traverse all potential scenarios in a single plane and develop a robust control system to respond to them, so that the computer vision tracking code can be implemented with knowledge of the required and expected functionality. With knowledge of the required logic, the robust control strategy achieved with the buggy can then be extended

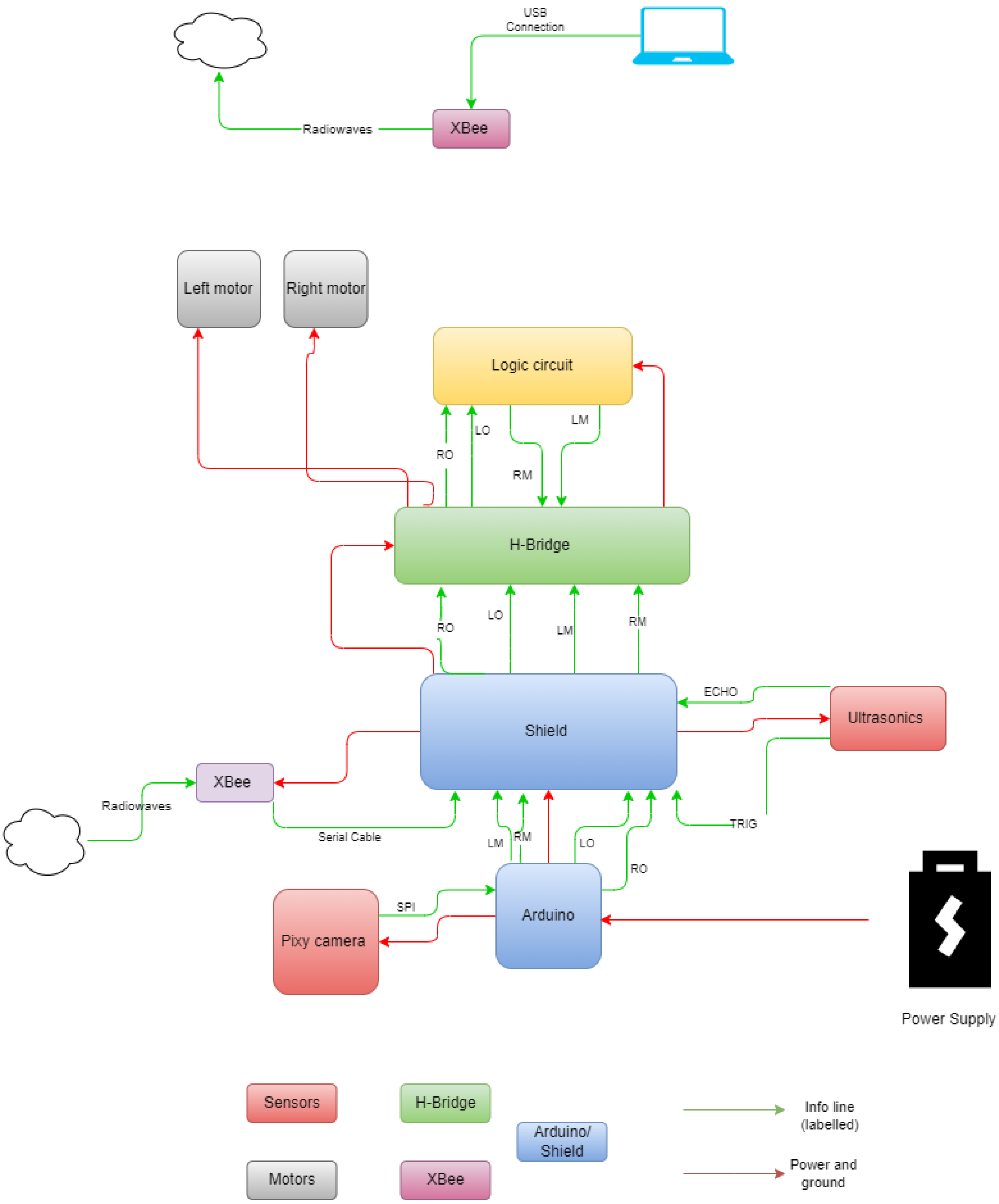


Figure 3.2: Block Diagram of Hardware

towards establishing control on a drone.

A key component of the system design was the obstacle detection and avoidance. The purpose of this was to ensure that the buggy is always safe in the event where an obstacle is encountered, or if the central controller instructs the buggy to go straight at max power towards the target or a wall. This can be implemented with additional sensors such as an ultrasonic sensors for the buggy, but these can be updated to lidar and radar as the system is scaled towards implementation on a drone.

Additional sensory information is necessary for unpredictable real-life conditions that the drone will be subject to. In the case of a large gust of wind or the turbulent airflow surrounding the turbine, it is vital to prevent damage to the drone that might occur by crashing into the target or an obstacle, or by falling to the ground.

3.2 Identification and Tracking

This section will outline the high level design considerations when developing a model to track and identify a moving target. To develop a highly robust control strategy, the identification and tracking of the systems target is vital. The aim of this system was to develop a set of instructions through computer visions to enable active target recognition and tracking.

The design considerations of this system were:

1. Target identification
2. Target recognition
   This was key requirement was that the system could distinguish between targets, as this is vital to ensure system reliability.
3. Track targets movement
4. Ensure the developed system is robust, reliable and accurate.

Each subsystem was developed considering a very simple case which models the movement of WTB. `Visual studio IDE` was used as the development platform for developing the computer vision system. All programming is in `Python` with tools provided by `OpenCV` used to advance the tracking system. `OpenCV-Python` was chosen as it provides a open source library of real-time python bindings designed to solve computer vision problems. It is a tool that can enhance image processing. The processing of all work was to be carried out in real-time to assist the system to make instant decisions.

A hierarchy arrangement of design was considered with each level having unique requirements. The aim was for the system to handle all situations identified by the buggy and develop a robust target tracking mechanism. The hierarchy designed comprises of:

1. Edge and Shape detection
2. Tracking using coloured targets
3. Target tracking

3.2.1 Edge and Shape Detection

The design requirement of this subsystem was to identify a shape through edge and contour detection. Once a shape has been identified, its area and centre coordinates are computed. The centre coordinates can then be used so that the system can track the target and also to assist landing coordinates to be computed. The area of each shape was calculated so that the system can distinguish between targets that are far away or very close.

3.2.2 Colour tracking

For colour tracking it was important to carry out all image processing in real-time. For modelling purposes a minimum of three colours were to be identified as this would simulate the motion of the three blades of a wind turbine.

The requirements were:

- Identify a target by its colour and place a marker to identify the target.
- Track target moving and display its centre coordinates and area
- Optimise robustness in the design so that it can be easily scaled.

3.2.3 Target tracking

When tracking the blades of a wind turbine, all targets will be of the same shape and colour. The aim was to develop a system to identify three targets of the same shape and size and distinguish between these targets based on a identifier such as a number or letter. The system must be able to handle occlusion: even in the event that two targets cross paths, each must be correctly classified.

These requirements were considered essential while implementing the functionality.

3.3 Unmanned Aerial Vehicle

The type of drones being considered for this project were quadcopters with an 'X' configuration and a front facing camera. There is a gargantuan range of consumer and commercial drones on the market today with different functionality and capabilities, selecting an appropriate drone which meets all the requirements is vital to the successful implementation of this system.

The requirements considered when selecting the drone were:

- Sufficiently small to be transportable and have high manoeuvrability
- Sufficiently large to carry a camera
- Contains a built-in open source auto pilot
- Ability to carry a battery of sufficient power for testing
- Ability to run TensorFlow/Pytorch in inference mode
- Ultrasonic Sensors (Not essential but could provide additional functionality).

3.3.1 Nazgul V5

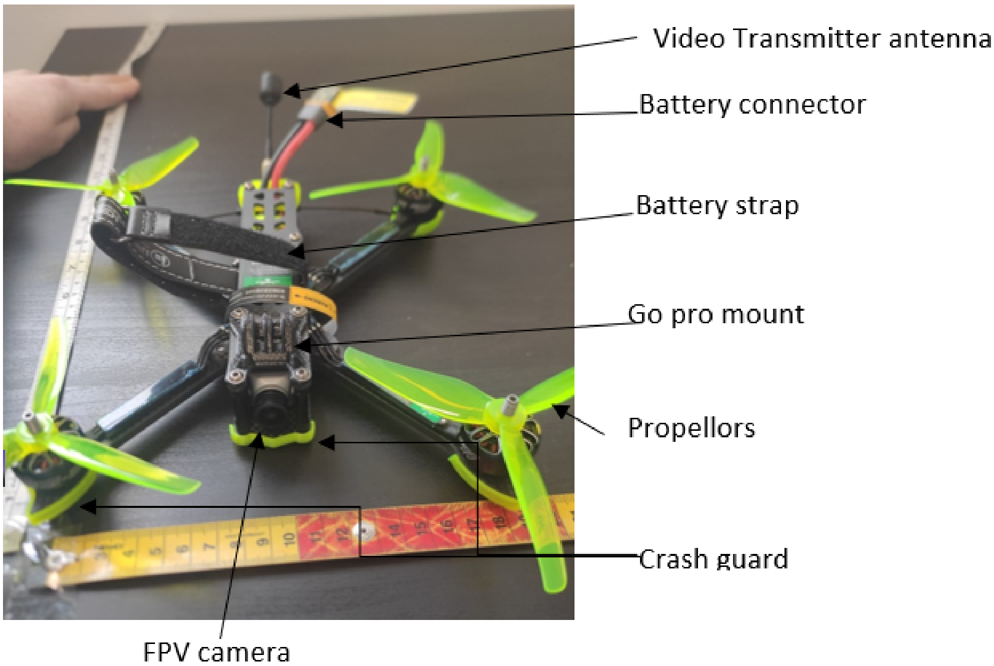


Figure 3.3: Components of the Nazgul 5

After extensive research and comparison, the Nazgul5 V2 5 inch quadcopter was selected. This is a freestyle quadcopter. The Nazgul has a pre-tuned setup, thus enabling relatively easy setup. This drone includes a succeX-E flight controller and video trasnmitter. The design of the Nazgul 5 is very durable with 5mm arm design, thermoplastic polyurethane (a class of plastic made from polyurethane) crash guards and covering on the arms to protect the wires. It is very light weighing only 393.4g (without its battery or additional camera). The Nazgul5 also includes a camera, an antenna for transmitting video, a GoPro camera mount and a battery strap. It was selected as it met all requirements, and is relatively easy and user-friendly to set up and test the system design.

There are two versions of the Nazgul5, the 6S and 4S. The 6S was selected as it offers more power, less voltage lag, longer flight time and it is more responsive and agile.

Further considerations were taken to ensure an appropriate battery was selected. The Nazgul requires a 22.2V 100C 6S battery. The maximum weight the drone can support for a battery, without reducing performance, is 200g and so a battery with 6 cells was selected.

4 Implementation

This section will explain how the design requirements were met. Firstly, how the functionality was implemented on the buggy . Then the algorithms will be outlined which were used for target tracking and identification.

4.1 The Buggy Implementation

This section will outline the implementation process on the buggy. The systematic approach was adopted and it was implemented to meet the design requirements. The buggy's safety was emphasised and control was valued above the buggy's speed.

4.1.1 Develop reliable and safe control

In order to develop control, the motors must be connected to the H-Bridge and the correct pins must be identified and connected. The buggys motor is a L293D H-bridge circuit. The H-Bridge powers the motors directly from the battery and enables the software to have complete control over the motor drives, controlling the speed, direction and electronic braking of each motor individually.

The enable pins are used to have the motor drivers ready to operate and by supplying a high signal these drivers are activated. The schematic of the L293D H-bridge was examined, and the motors were connected to the correct pins on the board. Once the pins were correctly identified, all motor control was achieved through Arduino.

The first significant implementation issue was encountered as the power of the left motor was significantly stronger than that of the right motor. A corrective factor was introduced to reduce the power of the left motor so that both motors were equally powered, and to ensure the buggy moved as expected. Reliable control necessitated getting the buggy to move forward, turn left, turn right and stop. Control signals were used to achieve reliable control.

Control Signal commands	
Control Signal	**Response**
MotorPowers	Sends power to the motors
LeftMotorNeg	Turns left wheel in reverse
RightMotorNeg	Turns right wheel in reverse
LeftMotorPos	Turns left wheel forward
RightMotorPos	Turns right wheel forward
CorrectiveFactor	A factor to account for the difference in motor power of the hardware
Turn	A factor applied during turning

Table 4.1: Control Signals

The following control functionality was implemented :

Forward: The buggy moves forward providing power to the positive motors.

Implementation:

- Check if the wheels are reversing. If they are, stop wheels reversing. Digital leftMotorNeg and rightMotorNeg are set to LOW
- Analog leftMotorPos $\times$ corrective factor set to left speed
- Analog rightMotorPos set to right speed

Stop: Buggy stops. In order to instantly stop the buggy reverse for 0.1 second before cutting power to all motors stopping the buggy.
Implementation:

- Digital leftMotorNeg and rightMotorNeg are set to HIGH
- Analog leftMotorPos and rightMotorPos are set to LOW
- Delay of 100 miliseconds
- Motorpower is set to zero.

Turn Left: The buggy turns left by providing the left motor with less power than the right.
Implementation:

- Digital leftMotorNeg and rightMotorNeg are set to LOW
- Analog leftMotorPost is further reduced (currently operating at 55% of the speed parameter as the right motor is damaged) and turn.
- Analog rightMotorPos set to motorPower.

Turn Right: The buggy turns right by providing the right motor with less power than the left.
Implementation:

- Digital leftMotorNeg and rightMotorNeg are set to LOW
- Analog leftMotorPositive set to motorPower
- Analog rightMotorPositive set to motorPower x Turn

Once solid control was established, it was essential that the buggy responded in an appropriate way to these simple control requests. Communication between the buggy and the controller is executed via the XBee.

4.1.2 XBee

The XBee is used to provide two-way serial communication between the buggy and the user interface. This wireless communication system was implemented to transmit control signals to provide user-override functionality. This system was also used to transmit updates of the buggy's activity status.

For the purpose of this research work two XBee devices were used. They were configured for wireless communication as is illustrated in Figure 3.2. The XBee models used are the XBee S1 and the XBee-PRO S1. One XBee is connected to the buggy via the Arduino as can be seen in Figure 3.1. This XBee communicates on the digital pins 0 (RX) and 1 (TX) while the other XBee is connected via a USB to the controller which is a laptop. On the Arduino board, the RX is the line which receives an incoming bitstream and transmits it to the Arduino, while the TX is the line which transmits an outgoing bitstream from the Arduino.

To implement the XBees, both are set to the same PAN ID and the baud rate of the port is initialised to be the same. Once this has been set up, the buggy can provide progress updates on the buggy's status and report via the TX stream. This is implemented simply by printing a message to the serial port for processing each time the buggy changes status, so that the user can see the status of the buggy in real-time.

This XBee was also used to transfer high level input commands and pass them to the low-level platform so they can be processed and translated into Arduino.

The accepted input commands are:

- ***Go and Start*** to start and stop the motors
- **Turn left** in the case where the buggy wants to avoid an obstacle, or the user wants to change the path. This causes the right wheel to rotate forward and the left wheel to rotate backwards to cause a sharp turn
- **Turn Right** similarly for the buggy to turn right the LeftMotorPos is provided power with rightMotorNeg given less power to result in a sharp turn.

Each time a messaged is received from the controller, it is processed and the correct function response is called. The message is processed as shown in the psuedo code1.

Status Report	
Status	**Meaning**
"Starting up"	The buggy is starting up
"Starting search"	System started and buggy is searching for target
"Object detected distance cm away"	Buggy stops as an obstacle has been detected, less than 15cm away
"Object removed"	Obstacle has been removed, buggy will search for target before continuing motion
"Target Identified "	System has identified target and stops the search
"Approaching target"	Buggy identifies and is driving towards target

Table 4.2: Status Updates

Algorithm 1 Translate high level commands into low-level Arduino commands

```
procedure READ AND TRANSLATE SERIAL MESSAGE
    top:
    if SerialmessageAvailable then
        message ← readString.
        for <all commands> do
            if message == command then
                Call function for command
                Serial print Status is command
                goto top.
```

4.1.3 Ultrasonics

This section outlines how the ultrasonics were used. The ultrasonic sensors were used to increase the reliability of the system and ultimately result in safe navigation. The buggy had one set of ultrasonic sensors positioned on the front of the buggy. This it is considered an effective safety mechanism to alert the buggy to potential obstacles and compute the distance from those obstacles.

The aim of using these sensors was to detect in real-time both static and dynamic obstacles, enabling the buggy to stop when necessary, increasing the flight control robustness of the device which is vital for safe control to prevent damage to the drone as well as to the obstacles encountered.

The sensor used is the HC-SR04 sensor. This sensor generates and receives sound waves using the piezoelectric effect.

The implementation process for the ultrasonic detection function was as follows :

- The echo pin is activated.

- A 10 microsecond high pulse is applied to the TRIG pin to send the initial ultrasonic pulse
- The module records the time between the TRIG signal being applied and the detection of a reflected wave by echo pin (which turns to high when an ultrasonic signal is being received). Based on this length of time, the distance to the obstacle can be calculated.

The ultrasonics were used for obstacle avoidance. If an obstacle within 15 cm is detected the Stop function stops the buggy. The buggy remains stopped until the object has been removed or relocates itself.

4.1.4 Pixy Camera

The Pixy camera was used to detect the target for the buggy. The target used was a high contrast pink target. The Pixy was used to identify the colour and provide an event driven command to the the logic, notifying it that the colour has been identified.

The quality of the pixy camera could be altered by focusing the camera. It was programmed with `PixyMon software` to recognize different objects by their colour signatures. The signature was set for a pink target in different lighting environments as the `PixyMon Software` is very sensitive to brightness and different light conditions. The PixyMon has additional settings such as brightness and contrast which were adjusted to optimise target visibility as well as setting the minimal number of pixels required for a block to be detected.

4.1.5 Implemented Logic

The micro-controller of the buggy was programmed in Arduino using the `Arduino IDE` and contains all the business logic of the system. Setup is the first function which is performed, and this runs at the beginning of each program.

The implementation of Setup is:

- Output message - *Setting up*
- Initialise the serial communication through the XBee
- Initialise the Pixy camera and begin gathering Pixy blocks

The system was setup to search, identify and approach the target.

Implementation of Search and Identify:

- Starts searching for the target by slowly rotating right and scanning the field of view for the target. Output message = *"Starting search"*
- Once a target is identified, stop search. Output message = "Target Identified"
- While the Pixy camera detects the target signature, drive towards the target. Output message = "Approaching target"
- If the Pixy detects colour and an object is detected, the pixel count is checked to see if it is above the threshold at target. It is assumed that if the camera sees most of the target while an obstacle is detected, it has reached the target. Output messeage = " Target found"
- If the Pixy camera does not detect the target signature, stop motion and search.

4.2 Target Identification and Tracking

This section outlines how a robust system was developed to track a moving target. The development of the model is clear as initially target identification implemented on images. This then progresses to tracking a coloured target from a live input. Finally a classifier is trained to track different targets. Each implementation was modelled for a simple scenario to ensure the logic of the applied system was correct.

To ultimately track the blades of a moving turbine, the hierarchy of implementation was:

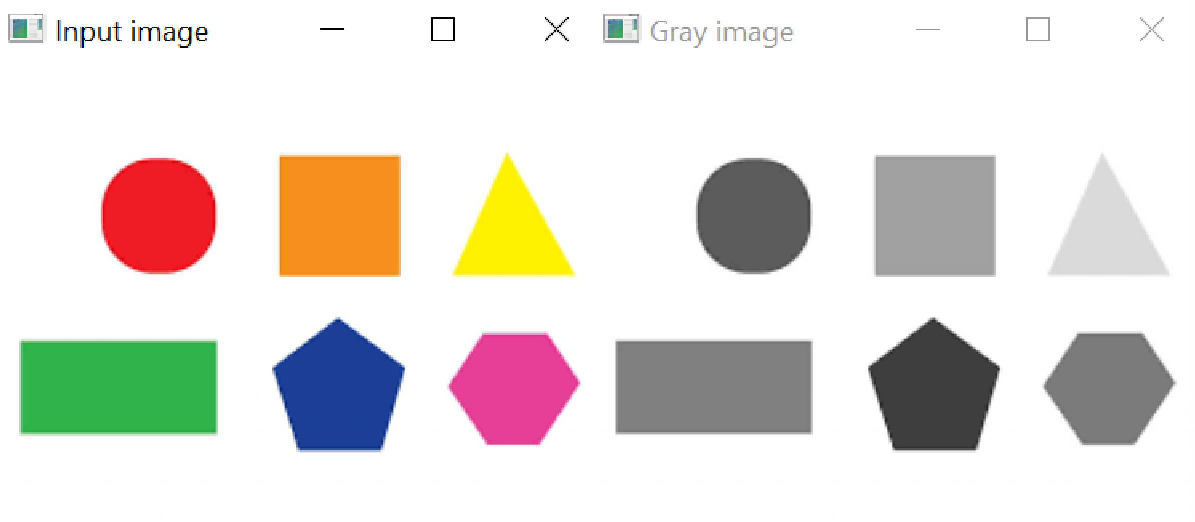
1. Identify and recognise different shapes
2. Track different coloured markers from a video stream and determine their co-ordinates
3. Distinguish between distant and close objects
4. Train a CNN to track different targets

To model tracking the blades of a wind turbine, three different targets were selected. A key design requirement was for the system to be able to distinguish between each target to ensure reliability of the system.

4.2.1 Edge and Shape Detection

This was implemented to model what the drone will see as it approaches the target. This can assist in identifying objects that the camera sees. It also assists in building a more robust model as it can successfully determine an objects shape and contours. This code was implemented to be tested on images of shapes. A simple function was used to determine the shape of the detected targets. The principle was if the edges and contour of the shape could be identified, the shape of the image could be determined. The functionality was implemented to detect the edges and contours of objects in an image and categorise the shape based on the findings from the edge detection.

Prior to finding contours the image was preprocessed. `OpenCV-Python` uses Blue Green Red (BGR) convention rather than the typical Red Green Blue (RGB). The BGR4.2a is converted to gray scale 4.2b. Colour can be considered as noise when detecting edges and this reduces the image to a 2D image, simplifying the processing. Gaussian blurring as shown in 4.3a is then applied to reduce the noise in the image and will improve the result of the edge detection. A kernel of size 7×7 was used to remove a large portion of noise. Image dilation4.3b was used to increase the thickness of the edges to assist identifying the shape. The Canny edge detector, an edge detection operator that uses a multi-stage algorithm to detect a wide range of edges in images, was the applied to identify the edges.



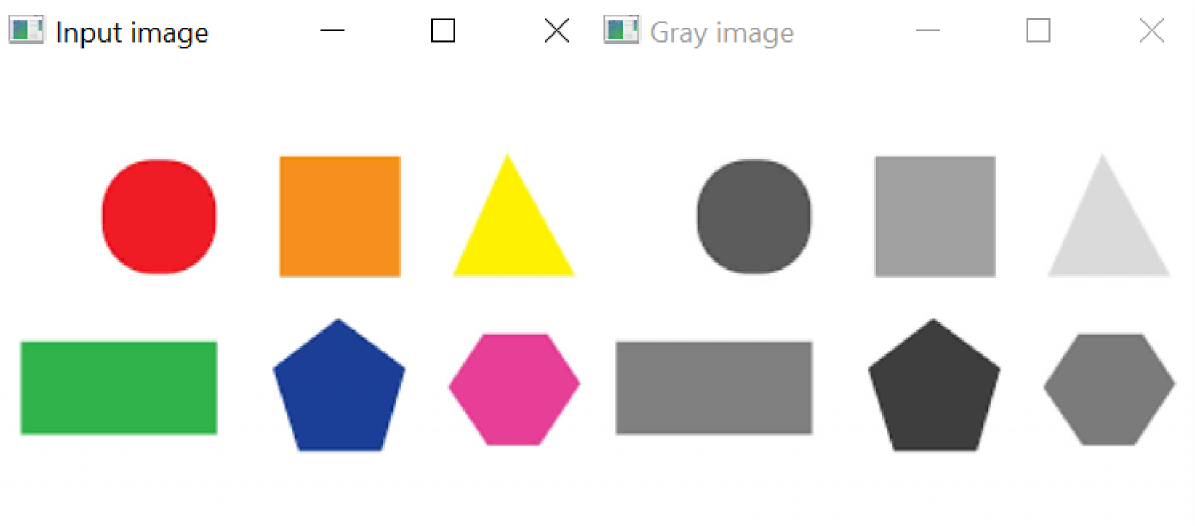
(a) Input Image (b) Gray Scale

Figure 4.1: Pre-Processing Stages one and two

Once the edges are detected, the extreme contours were found using the `OpenCV-Python` Find Contours functionality. The area of this contour can then be computed. The area is computed to provide a more reliable detection algorithm, as a threshold of minimum area can be set to ensure that noise is not identified. The contours of each image are then drawn in blue and displayed as can be seen in 4.4b.

Once the contours have been identified, the curve length is calculated so the corners of the shape can be approximated. The corners of each shape can then be determined, and based on the number of coordinates the shape can be identified. This was implemented for the test shapes as seen in 4.4b which were circle, square, triangle, rectangle, pentagon and hexagon.

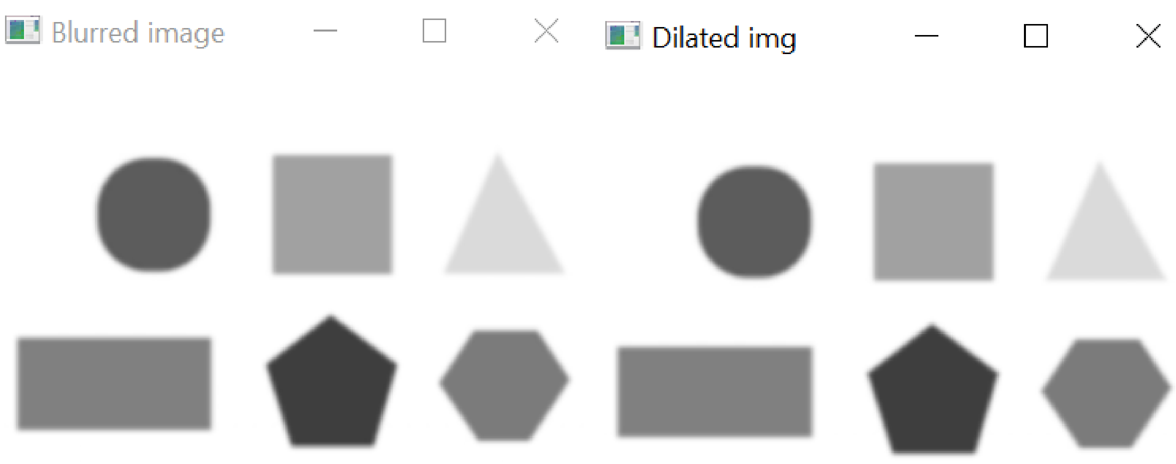
The stages of implementation are illustrated step by step with the images:



(a) Input Image

(b) Gray Scale

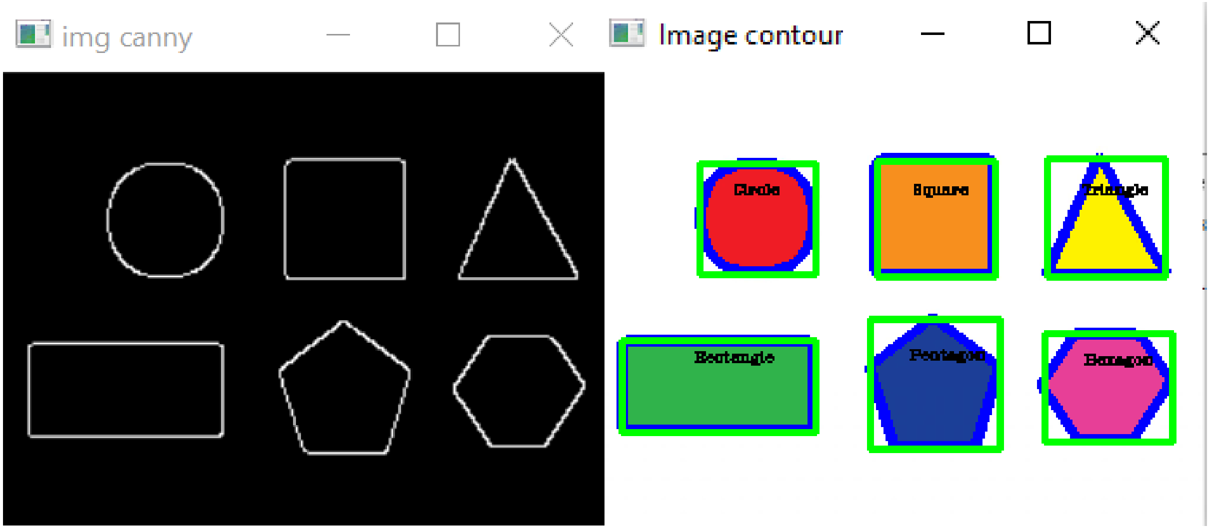
Figure 4.2: Pre-Processing Stages one and two



(a) Blurred Image

(b) Dilated Image

Figure 4.3: Pre-processing Stage three and four



(a) Edges detection Image (b) Final output image

Figure 4.4: Edge and Contour detected

4.2.2 Tracking targets using colours

The aim of this component was to:

1. Identify a target based on its colour.
2. Distinguish between each targets based on colours and place an identifier on each target.
3. Track the movement of each target by displaying its coordinates.
4. Compute the area of each target to determine which is closest

Identifying colour

This code was implemented using `Visual Studio IDE` and `OpenCV-Python`. Three targets of colour green, yellow and orange of the same size were used.

To identify the colour of each target colour masking was used. The mask applied to each target was found through trial and error. This was achieved by placing the target in front of the webcam. Colour space of the image was converted from Blue Green Red (BGR) to Hue Saturation Value (HSV). The hue, saturation and value limits for each colour were then selected for each target. These values were found through trial and error using track bars to find the optimal minimum and maximum values of hue, saturation and value for each colour. This range was recorded to represent the target the mask required to identify the target. This process is illustrated in Figure 4.5.

In Figure 4.5 the first image is the input image as seen from the video stream. The second image is the mask applied to identify only the colour of interest. The mask consists of the higher and lower limits of hue, saturation and value. The third image is the output after the mask has been applied to only show the colour of interest.

Figure 4.5: Colour Masking for Green target

Locate target

The next process was to locate the target in the image. To do this the contours of the target were identified. `OpenCV` in-built functionality was then used to determine the area of the contour. A threshold was applied for the minimum area of each target in order to eliminate noise. The area of the identified target was computed and displayed to distinguish between objects which are close or far away. To track the target, the coordinates of the target were identified and displayed. This was done by finding the x, y, width and height coordinates of the bounding box of the target. The centre of each target could then be computed and this was displayed1.

$$(\frac{x + width}{2}), (\frac{y + height}{2})$$

(1)

A marker was then place at the centre of the top of the target, and these coordinates were returned. The area of the target was also computed as the area of the contoured object to help determine the size. This can help determine if the target is close or far away.

An marker identifier is displayed on each bounding box by placing a coloured circle the same colour as the target on each target.

4.2.3 CNN model

The aim of this model was to dynamically track the motion of targets and distinguish between three targets. Through research of previous work it was identified that a trained model may provide more reliable functionality(32). As the aim was to develop a reliable and robust model, a Convolutional Neural Network was trained to track three moving targets. Using `Keras` and `Tensorflow` to implement this model.

The model being developed is a supervised learning algorithm which aims to learn to predict target values from a labelled set of data. While there are large labelled datasets available, the quality of the dataset is crucial to the models performance. When training a classification algorithm a labelled dataset is required, it is vital that the algorithm is trained on representative data and that the dataset is of a sufficient size for the algorithm to learn about the features.

Otherwise the classifier will be of poor quality as it will not have learned sufficient information from the data to generalise well. For this reason a dataset was created.

The dataset used for training the model was collected through video stream. The video is read frame by frame, and each frame is saved. Then using `Labellmg tool` each frame was annotated. The dataset has four classes, one for each target and an additional class was created called noise to reduce the wrong detections. When training the model a Train-Test split of 80-20 was used, so a true representation of the algorithm's performance could be evaluated.

A limitation of this research work is that it is difficult to get access to a large database of aerial images. Although the annotation of images can be considered a time consuming process, it can ensure that the database is a true representation of the scenario it is representing.

A CNN classifier is trained on these frames. The classifier architecture was adjusted to minimise the loss and maximise the accuracy by adjusting the learning rate, activation function and optimizer used. To reduce overfitting, dropout was applied.

As there are four classifications, multi-object detection was used to identify each target. To perform the multi object detection, the selective search (SS) algorithm was used. In selective search the frame is split into multiple Regions Of Interest (ROI). Each region is then classified using the trained classifier. If the confidence of the classification is above 90%, detected box is added to the list of proposed boxes.

Non Maximum Suppression (NMS) is used to filter the number of detected boxes and select the most appropriate bounding box for each target. The function's inputs are a list of proposed boxes, the corresponding confidence provided by the trained classifier and a overlap threshold (of 0.09). The function returns a filtered list of boxes which were selected.

The implementation of NMS is:

1. Select the box with the highest confidence score, remove it from the list of proposals and add it to the filtered list.
   ***Note:** *All confidence values are above 90% as they have been previously filtered.*
2. Compare the overlap or Intersection over Union (IOU) of this box with all other proposals in the list of proposals. The degree of overlap is compared with the threshold. If it is above the threshold the box is removed from the list of proposals.
3. Then the proposal with the next highest confidence is taken from the list of proposals. It is removed from the list of proposals and added to the list of filtered boxes (as in step 1)
4. Steps 2 and 3 are then repeated until the list of proposals is empty.

After NMS has been applied to subdue the overlapping detections, the final output of the NMS was displayed.

4.3 UAV implementation

The aim of this subsystem was to then implement the functionality achieved with the buggy and target tracking system on the drone.

This would be implemented in stages as shown:

- Test safe control of the drone in a controlled environment
  To test safe control this would be achieved by removing the propellers from the device and setting up the drone for testing. The motor response could be check to ensure that the drone will move as expected prior to testing in a outdoor environment. Simple testing at a limited power capacity but would be tested outside to ensure the drone moves as expected by attaching the drone to a piece of string and testing the system commands.
- Get the drone to fly autonomously and search for a stationary target in the horizontal plane. On identification, approach and land on the stationary target.
  Once reliable control has been established the target tracking system will be tested by getting the drone to identify a stationary target that is located in a plane. An experimental setup for this would consist of three large targets located at the end of the field. The drone will approach them identify the target and safely land on it.
- Get the drone to autonomously fly and identify a target moving in the horizontal plane. On identification, get the drone to track the targets movement prior to landing.
  The experimental setup to achieve this would be to setup a car moving in a horizontal plane with a target attached to the roof. The drone would then identify the target and track the cars motion in the horizontal plane before landing on the moving target.

Once this functionality has been tested this will provide confidence in the system design, demonstrating that it can safely and reliably identify and track from active a moving target.

5 Evaluation

In this section the experimental findings will be analysed and evaluated. Firstly the functionality of the buggy will be examined before progressing to the methods implemented to dynamically track a target.

5.1 Buggy

This section will look at the functionality achieved with the buggy, and the results obtained from the evaluation of the Pixy Camera and Ultrasonic sensors. To measure the accuracy of the results from both, two experiments were carried out.

5.1.1 Experiment One: Pixy Camera

To evaluate the Pixy camera's Field of View, and to measure the reliability and accuracy of the results, an experiment was set up as shown in Figure 5.1. It was important to establish the Pixy camera's FOV to determine where a target could be identified. To examine the FOV, the Pixy camera was placed at 0 cm and a target was moved back as far as 400 cm in the x-axis, and also moved across the y-axis to establish the range (vertical scope).

This experiment demonstrated that while the Pixy camera does have a very large FOV, the limiting factor is that the quality of the video stream from the camera is very susceptible to lighting variations. The Pixy camera was found to have a significant lack of contrast and could only detect high contrast targets.

As a camera, it was found not to be very reliable. To overcome this, a high contrast pink target was used. Different signatures were set for the target in different lighting conditions and at different distances to ensure the target would be identified. This is illustrated in Figure 5.2. While this design does achieve the necessary functionality for the buggy, this is not a reliable control strategy for a production environment. This emphasised the need for a more reliable target tracking model to ensure the target was always identified.

5.1.2 Experiment Two: Utrasonic Sensors

To evaluate the Field of View of the ultrasonic sensors, to determine the reliability of the sensors and to provide confidence in the selection, an experiment was set up as shown in Figure 5.3.

The sensors' performance was measured by gradually modifying the distance between the sensors and an obstacle placed in front. Using the XBee, the distance was measured and

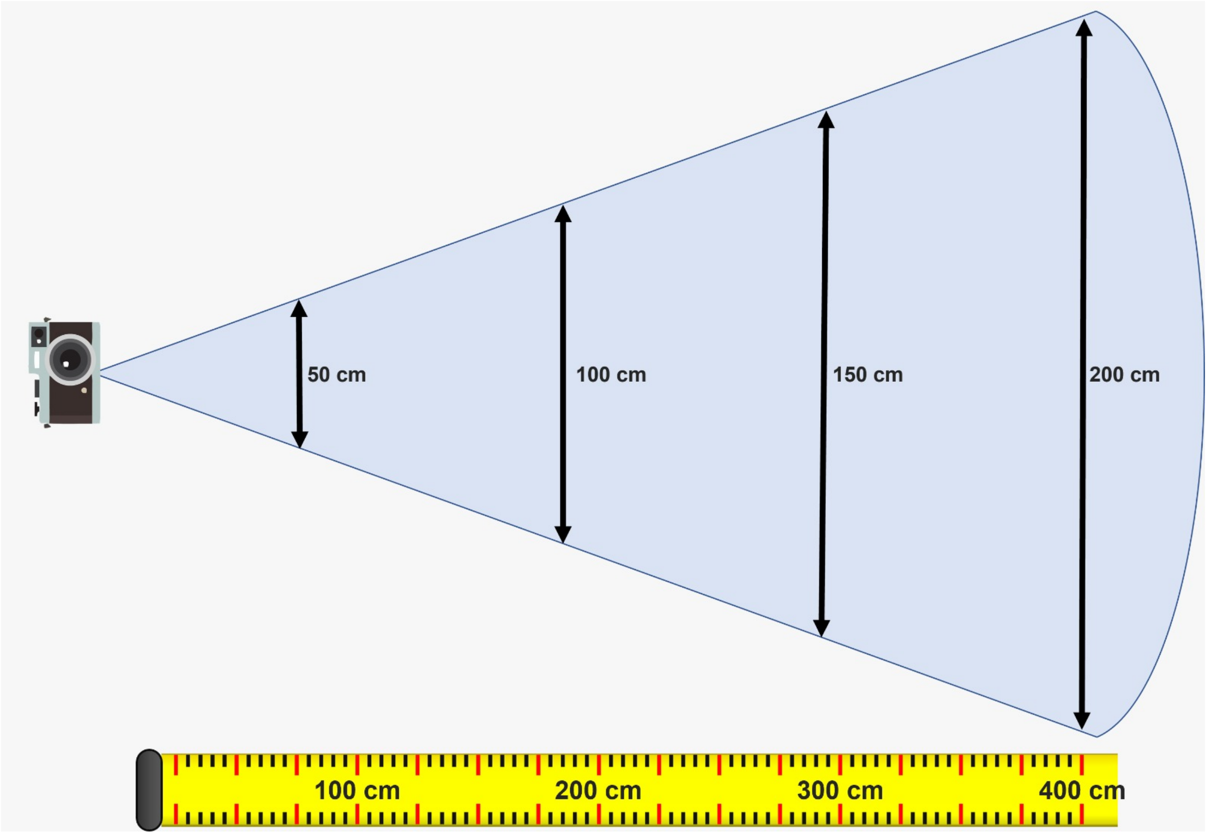


Figure 5.1: Experimental Setup: Pixy Camera FOV

(a) Target Identification close proximity

(b) Target identification at a distance

Figure 5.2: Pixy camera POV

displayed on the controller interface. The distance was increased gradually from 0 cm to 400 cm as shown in Figure 5.3.

This experiment revealed that the ultrasonic sensor can detect an obstacle up to 400 cm in real-time. However, it also identified that the range (vertical scope) of the ultrasonic sensor was very low, with a maximum range of 20 cm being recorded. The ultrasonic sensors can only detect an object if it is near the centre of the visual field. To improve the reliability of the results provided by the sensors, additional sensors could be added to expand the field of view of the buggy enabling it to detect objects in its peripheral vision.

This demonstrated that while ultrasonics can be used to detect obstacles, a single ultrasonic sensor should not be used independently. It also provides insight into the control design for

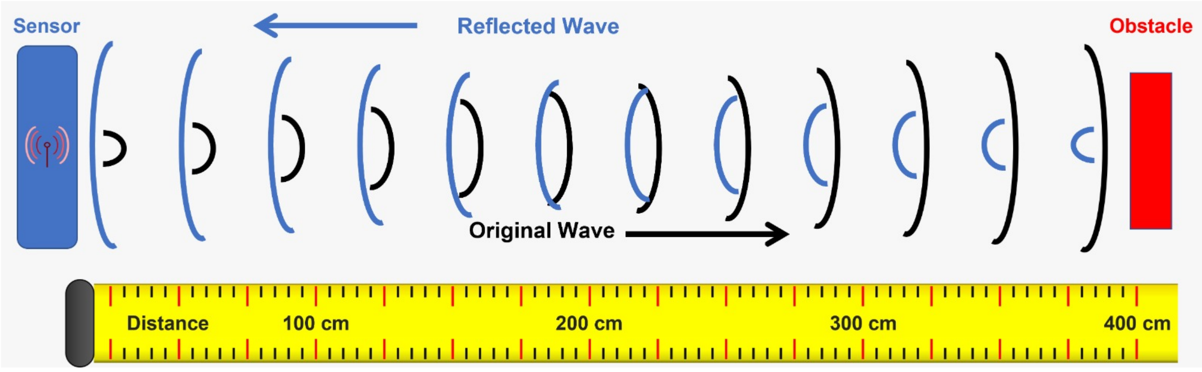


Figure 5.3: Experimental Setup: Ultrasonic sensor

the drone: an ultrasonic sensor should not be used independently for obstacle detection but should instead be used with computer vision, to increase the reliability of the computer vision system to ensure a collision does not occur.

A key consideration to this system design was to ensure that safe control was achieved. To ensure this, the ultrasonic sensor is used in combination with the Pixy camera as seen in Figure 5.4 to provide reliable obstacle avoidance and target detection.

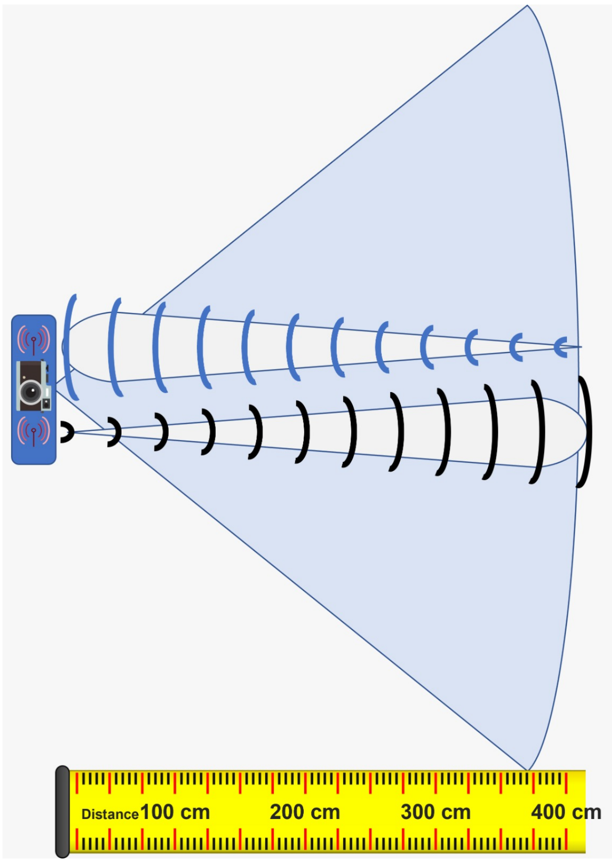


Figure 5.4: FOV of buggy

5.1.3 Experiment: Buggy control

All design requirements for the buggy were met, namely:

1. Reliable control
2. Search and identification of a target
3. Obstacle avoidance
4. Serial communication.

In the test environment, the buggy successfully identified and approached a target. The buggy responded immediately to commands provided by the user and the obstacle detection alerted the buggy to approaching obstacles so that it could respond appropriately. The buggy also provided status updates to the user interface. The development of this system provided a deep understanding of the functionality required to identify and safely approach a target.

5.2 Edge and shape detection

To detect edges and shapes, a series of experiments were completed, with increasing complexity of images.

5.2.1 Experiment 1: Simple shape detection

The algorithm was modelled and initially tested on a single image of 6 shapes (circle, square, triangle, rectangle, pentagon and hexagon). The model performed well, and correctly detected the edges of each shape and then classified each image based on the number of corners.

The key achievement of this model was to detect edges. This is a powerful tool in object detection as once the edges which been detected, the contour of the object can be found. This enables the program to locate a target and compute its coordinates. The area of each object can also be calculated so that a distance measure can be considered based on how large ('relatively close') or small ('far away') an object is. The classification method used is very simple and specific for this model but it demonstrates that targets can be identified and recognised on shape alone.

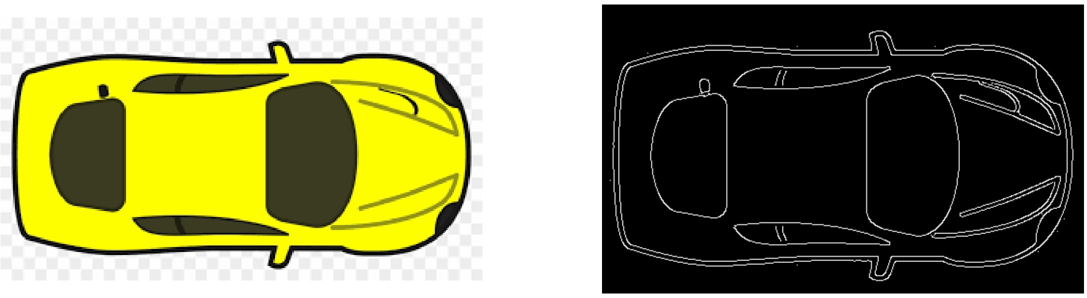
To further analyse this method, experiments were carried out using different, more complex and realistic images to evaluate if this model is reliable and can be used to correctly identify the edges of targets.

5.2.2 Experiment 2: Edge detection of a car

The same edge detection code was then applied to detect the contour of a car from a aerial viewpoint. This case was considered as it would develop this work towards the goal. Initially the aim was to track the motion of a target (such as a car) in the horizontal plane. It was considered important to test the implementation in a more realistic scenario. The scalability of the model was also tested in this experiment to ensure that it worked for more that just the test case used when implementing.

The modifications made to the code were as follows:

- Remove classification component based on the number of corners as no longer considering shapes.
- Increase the size of the minimum contour so only the outermost edge is detected.
- Display the calculated area and centre coordinates.



(a) Input Image of Car(39)

(b) Edge detection of Car

Figure 5.5: Pre-processing of Car

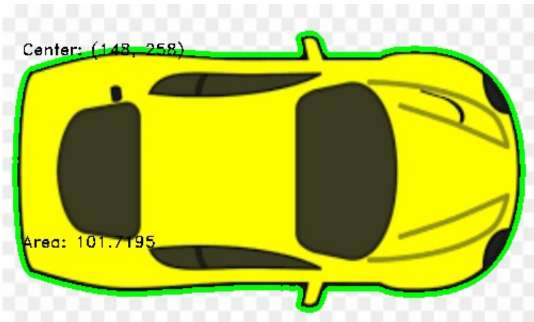


Figure 5.6: Car detected

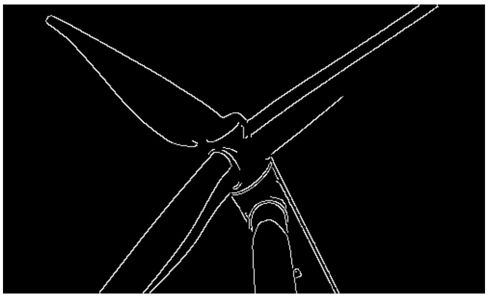
As seen in Figure 5.6, the perimeter of the car was correctly identified and outlined, indicating the designed method can be used for more complex situations. There is also little noise influencing the detection of the perimeter. The method used to reduce noise was to use a threshold to filter the size of the calculated areas. This is evident when Figure 5.6 is compared to Figure 5.7 where the area of the largest perimeter is the only one outlined in the final image.

This demonstrates that the method implemented was adaptable and reliable.

5.2.3 Experiment 3: Edge Detection of a wind turbine

This experiment tested the functionality on the more realistic case of a wind turbine. The test image used was subject to background noise, and the minimum threshold of the area was reduced in this case to ensure each that each blade of the wind turbine could be correctly identified. As the drone approaches the target, it will be required to detect the target from all points of view, not only when the target is near the centre of the visual field. This experiment also measures how robust the edge detection method is.

(a) Input Image of Wind Turbine(40)



(b) Edge detection of WT

Figure 5.7: Pre-processing of WT

Figure 5.8: Edge detection of Blades

The input image can be seen in Figure 5.7a. The edge detection is not as smooth and continuous as seen in Figure 5.5b due to the curvature of the blade. To reduce the discontinuities, the dilation factor was increased to try identify more edge pixels and the Canny threshold was changed. This did not reduce the discontinuities. Components of each blade are identified but the entire perimeter is not outlined due to discontinuities. This demonstrates that the edge detection method is dynamic as it can identify edges in more complex shapes. However the perimeter is not always correctly identified as seen in Figure 5.8, in cases where only parts of the blades are detected.

The experimental findings illustrate that the algorithm developed successfully identifies the edges of a target object. The algorithm is dynamic and scalable as it works on different images. However the perimeter is not always correctly identified as seen in Figure 5.8 in cases where only parts of the blades are detected. Further research is required to improve the target identification algorithm to ensure a robust target identification and recognition system is developed.

5.3 Tracking of coloured targets

The tracking and identification of targets based on colour was tested using three coloured targets of the same size.

5.3.1 Experiment 1: Tracking coloured targets

The input frames were taken from a video stream to model real-time identification and classification.

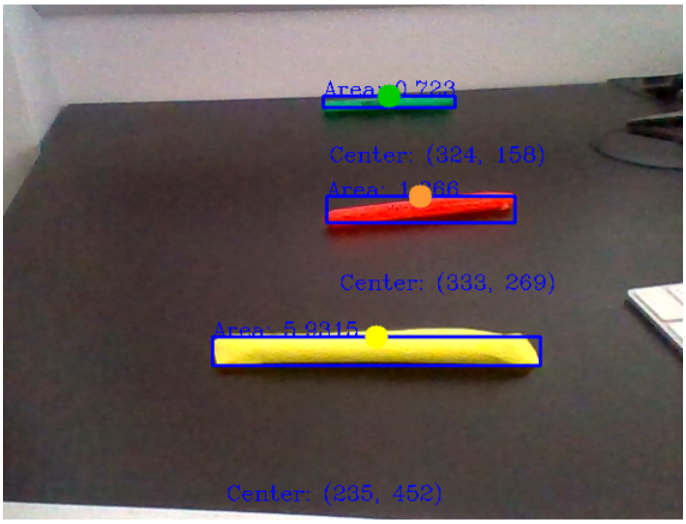


Figure 5.9: Coloured targets Identification

Figure 5.9 illustrates the performance. Each target is identified and a coloured marker (a circle) is placed on each target to reflect the colour which is observed.

Each target is of the same size. Based on the computed area, it demonstrates that the yellow target is closest while the green target is furthest away.

This system was found to have perform as expected, correctly identifying each target and tracking its motion during testing. Experiments demonstrated that the system was found to instantly identify and track each target, reflecting no lag in the identification process as required in order to achieve reliable control.

5.4 Target tracking

To further enhance the model, three identical targets of the same shape and size - distinguished by an identifier - were tracked.

5.4.1 Experiment 1: Tracking moving targets

Three yellow targets were used which were distinguished by hand drawn Roman numerals. To create the dataset, a video of these targets moving was recorded live, where each frame was captured. 21 frames were annotated from the complete dataset collection of 1,800 frames. Each target was labelled as T1, T2, T3, and an additional class called Noise.

This dataset was used to train a CNN classifier. To optimise the classifiers performance, a graph was plotted of epochs versus accuracy and loss, where the loss function used was categorical cross-entropy.

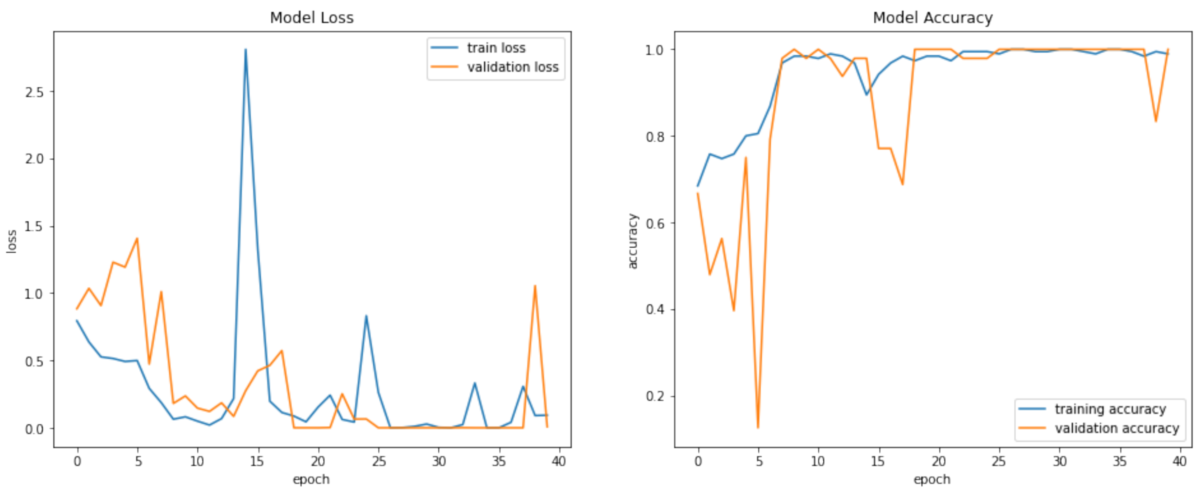


Figure 5.10: Performance evaluation, Adam optimizer and a learning rate of 1×10^{-3}

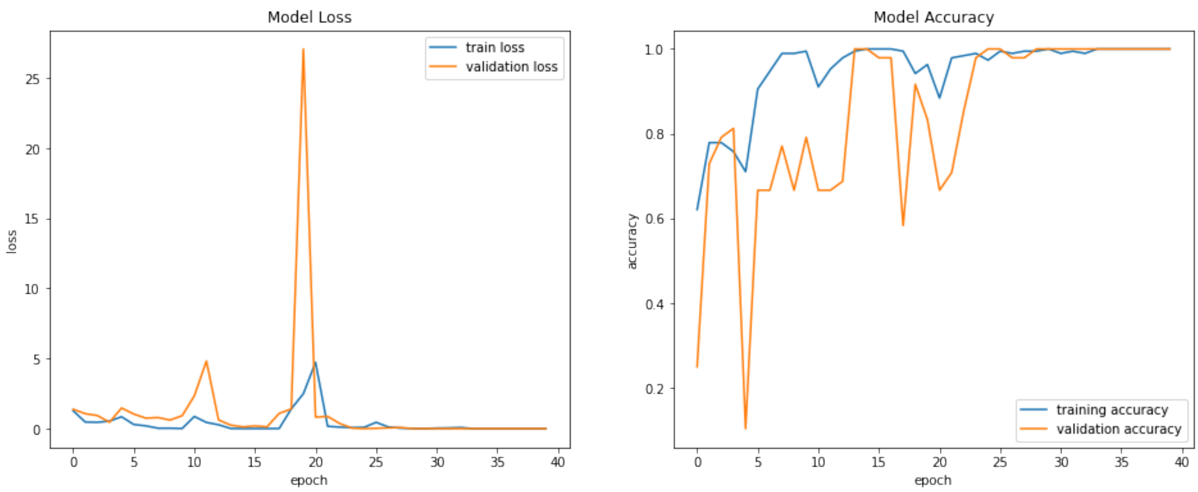


Figure 5.11: Performance evaluation, Nadam optimizer and a learning rate of 1×10^{-3}

As Figures (5.11, 5.10 and 5.12) illustrate, each model was trained over 40 epochs using a batch size of 8. To indicate that the model was performing well, high accuracy and low

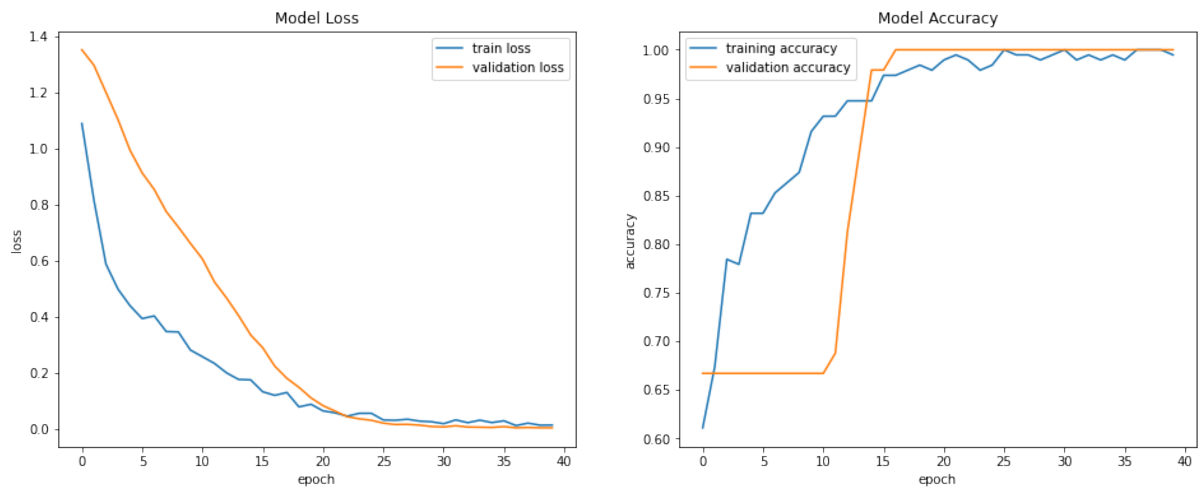
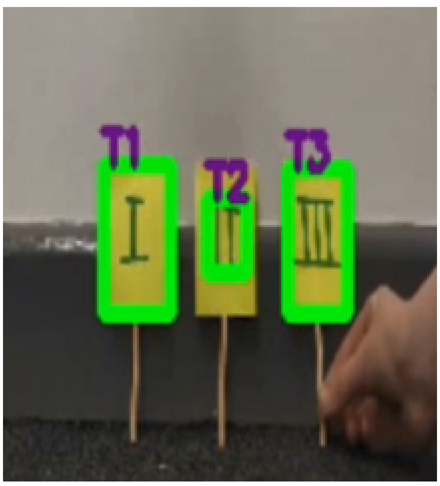


Figure 5.12: Performance evaluation: Adam optimizer and a learning rate of 1×10^{-4}

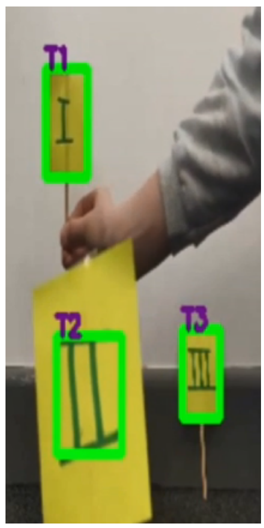
loss was required. Figure 5.10 demonstrates that the learning rate is too high and a optimal solution is not found. The loss does not continually decrease with increasing epoch but randomly oscillates. This reflects the model has not converged and would result in unreliable results. The accuracy graph for the Nadam optimiser Figure 5.11 appears to converge after 30 epochs. However, although the loss is consistent, it is quite high indicating that the model has not learned enough about the data.

Figure 5.12 demonstrates optimal performance. The loss of the validation model is lower than that of the test data, demonstrating that the model has learned to generalise the data. The accuracy graph also reflects this and convergence is observed after 15 epochs. The model performs better on the validation set than on the training data. Considering that a small dataset was used for training, the model performed very well.

The performance was then tested on a video input.



(a) Target Identification

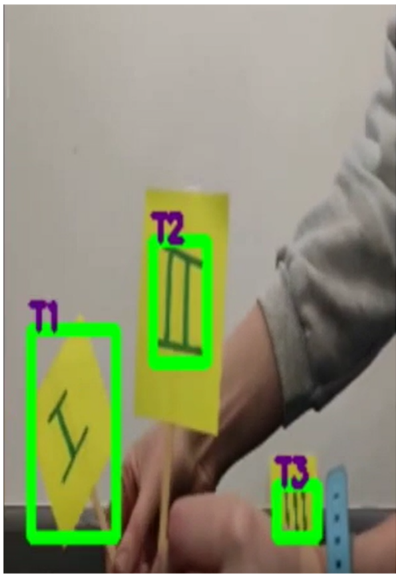


(b) Identification with background noise

Figure 5.13: Performance during testing

The classifier worked very well during testing as can be seen in Figure 5.13. Each target is correctly identified and its motion is tracked. Figure 5.13b demonstrates that the system handles noise very well, as the targets are partially obstructed by obstacles.

(a) Target two identification



(b) Re-identification of targets 2 and 3

Figure 5.14: Occlusion handling

A key design consideration was that the model should handle occlusion, as this is essential for a robust classification model. This has successfully handled occlusion and target re-identification if a target leaves the cameras FOV. This can be seen in Figure 5.19. In Figure 5.14a target T2 is covering T1, and T3 is hidden by an obstacle. When T1 and T3 return into the field of view, they are re-identified.

On reflection the model works very well. To improve the model, a larger dataset could be annotated, and the model's performance under different lighting conditions could be tested. When creating the dataset every frame is captured and saved which results in a large amount of data being collected. To reduce the dataset size, a frame at one second or two second intervals could be captured to avoid multiple identical frames and reduce the memory required during dataset creation.

5.4.2 Experiment 2: Tracking moving cars

Figure 5.15: Experimental set up

To test this model further, a more realistic experiment was set up as shown in Figure 5.15 using three model cars of colour red, green and blue. Each car was attached to a wooden stick so that it could be moved to simulate a car's movement. To extend the functionality towards drone control, the aim was to track a target moving in the horizontal plane which this experiment simulated. The camera was set up with an aerial view to make the experiment more realistic.

The four classes used were:

1. **Target one:** Red car
2. **Target two:** Blue car
3. **Target three:** Green car
4. **Noise** Background noise from the video

Data was created from a video dividing it into frames and annotating selected frames. Initially 50 frames were selected and annotated for the dataset. The graph of epochs versus loss in Figure 5.16 demonstrates overfitting as the validation loss initially decreases but with increasing epochs it begins to increase. The epochs versus accuracy graph also demonstrates that overfitting is occurring as the accuracy on the trained model is higher than that of the validation accuracy. The results in Figure 5.16 demonstrate that the learning rate is too high, and this is reflected in the oscillating graph. The dropout applied was increased in an attempt to reduce overfitting, but that did not have a significant impact and so it was evident that a larger dataset was required.

To increase performance of the model, the size of the dataset was increased and 82 frames were annotated. On initial testing with a dataset of size 50, the dropout was increased in an

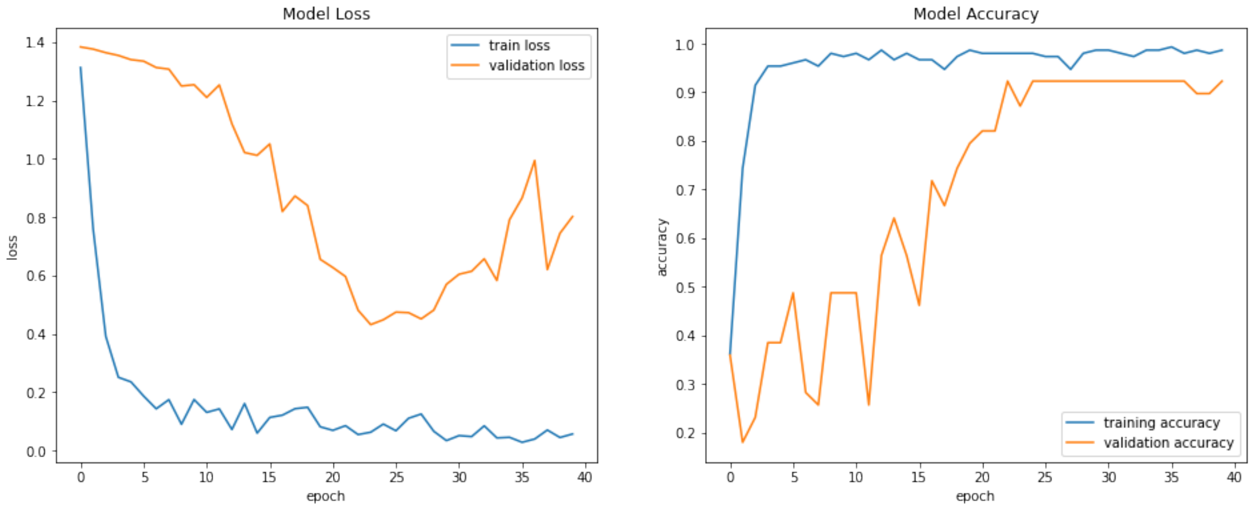


Figure 5.16: Performance evaluation, dataset 50 images, Nadam optimizer and a learning rate of 1×10^{-4}

attempt to reduce overfitting. When training with the larger dataset the original architecture used in Experiment One was maintained. A learning rate of 1×10^{-5} was applied, and the Adam optimizer was used when training the model.

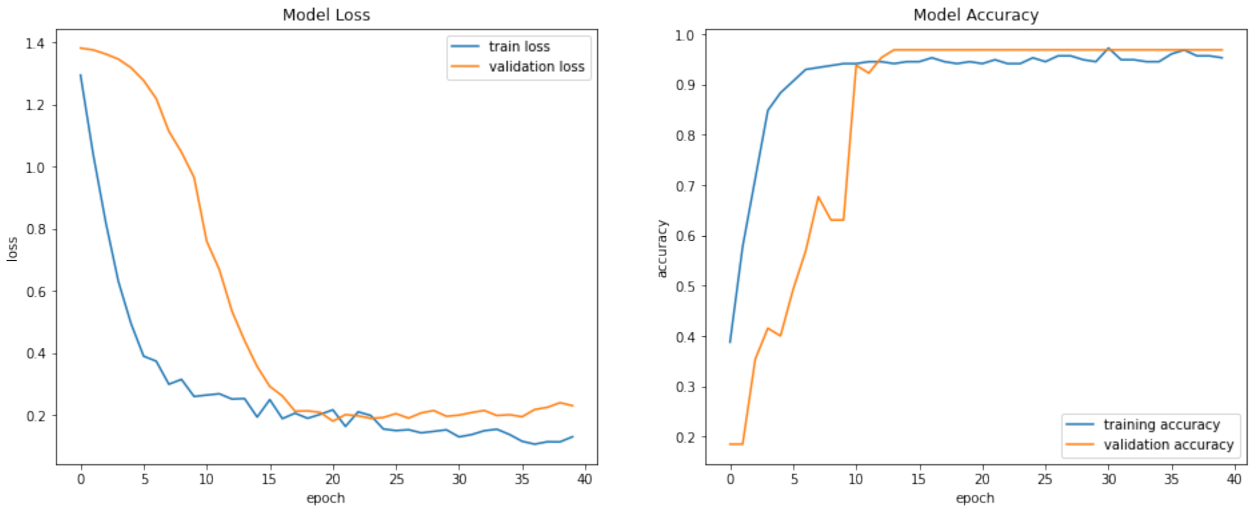
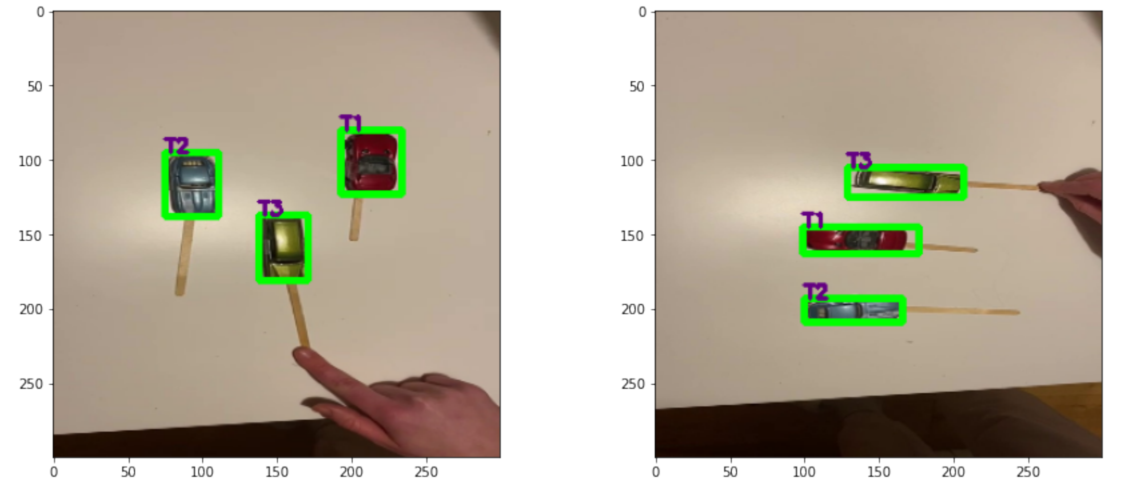


Figure 5.17: Performance evaluation, dataset 82 images, Adam optimizer and a learning rate of 1×10^{-5}

As seen in Figure 5.17, a validation accuracy of 96.92% was achieved, demonstrating that the model has learned to identify each target. The validation loss has decreased significantly and is much closer to the training loss reflecting that the model is a better fit.

During testing the model was found to identify targets in both a horizontal and vertical orientation. This is reflected in Figure 5.18. The model successfully tracks target motion in both directions, illustrating that the model is robust to different orientations.

The model also successfully recognises targets if partial occlusion occurs. This is demonstrated in Figure 5.19. This also reflects the robustness of the design as all targets are identified when noise is introduced by an obstacle in the frame.



(a) Target Identification vertical orientation

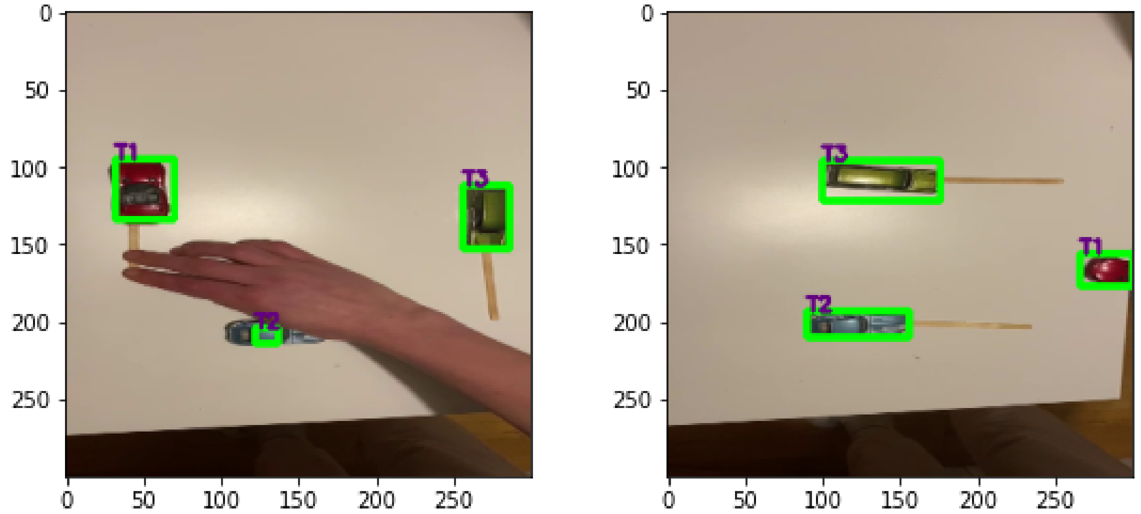
(b) Target Identification horizontal orientation

Figure 5.18: Target Identification in both orientation

Finally, in the event where a target is removed and reintroduced into the frame, the model re-identifies the target. To demonstrate this, Figure 5.20 shows frame 1762 and 1778, in which target one has been removed from the FOV and then reintroduced.

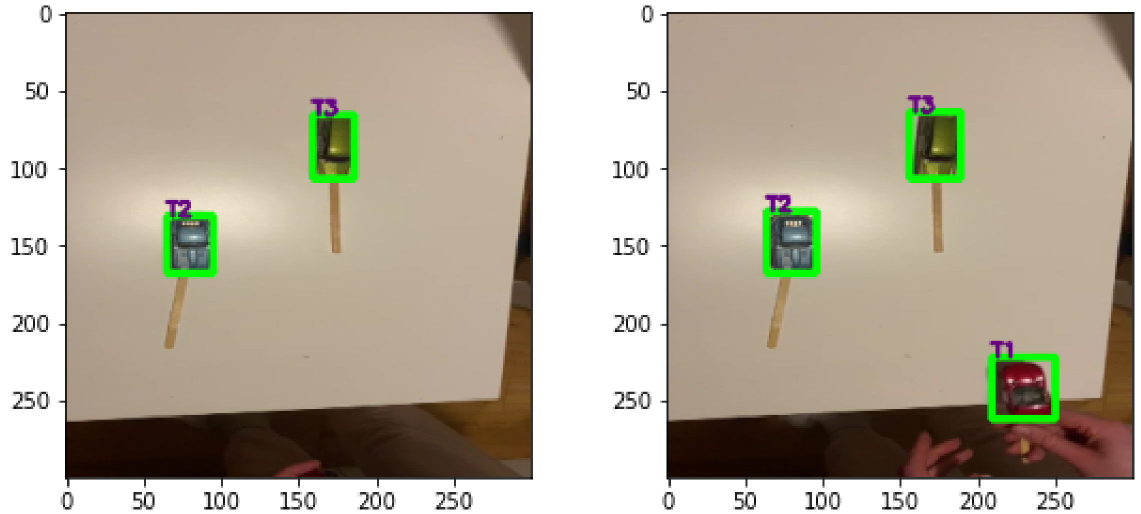
The model successfully tracks and identifies an object in the horizontal plane. Testing on the video stream revealed that the model is robust, reliable and accurate in the simulated environment. This experiment provides confidence that the developed model is dynamic to more complex situations providing that it is trained on a sufficiently large dataset.

To progress this research work towards UAV autonomous control, target tracking must be achieved in the horizontal plane prior to tracking a target in the vertical plane. Experiment 1 simulates tracking with a first person perspective while Experiment 2 simulates tracking from an aerial perspective, and both demonstrate reliable and accurate tracking and identification.



(a) T2 Identification in the event of partial occlusion (b) T1 Identification in the event of partial occlusion

Figure 5.19: Target Identification in the event of occlusion



(a) Frame 1762 (b) Frame 1778

Figure 5.20: Consecutive frames to demonstrate re-identification

5.5 Summary of Results

This chapter details the experiments conducted in the four key components of the project. It has shown that all four areas were addressed effectively, and high performing algorithms implemented.

The use of Pixy cameras and ultrasonic sensors were investigated and encouraging results obtained for the future deployment on a drone. Buggy control was achieved to identify, locate and approach a target. A suite of tools for identification of targets was developed, using edge and shape identification, colour tracking and tracking of moving targets.

6 Conclusion

This chapter is a summary of the main findings and contributions of this work.

6.1 Overview

The primary objective of this research work was to develop a robust,reliable control system to enable a drone to autonomously track and identify a moving target. This research was categorised into two main areas, autonomous control and computer vision, both were successfully implemented. A system was developed and tested to ensure its reliability, results demonstrate that a validation accuracy of 92.16% was achieved for target identification,recognition and tracking.

Autonomous control was established in a simplified environment, this was achieved with the buggy. The final steps in this project would involve extending the functionality of the buggy to the drone, this is addressed in the future work section 6.2.

The key requirements as identified in the design chapter are:

- Establish autonomous control
- Identify the target
- Search for and move towards the target
- Avoid obstacles
- Implement a user override functionality
- Distinguish between targets
- Track targets

All were successfully implemented with a buggy equipped with an XBee controller, camera, ultrasonic sensors and Arduino.

A robust target tracking system was developed which handled occlusion and re-identification. Its functionality was simulated in a realistic environment and it was found to successfully identify and track a targets motion in both the vertical and horizontal plane.

6.2 Future Work

The future work of stems from the limitations of this work. If an additional two months were provided project and providing a battery and suitable charger were procured, the future work would be centred around implementing the functionality achieved thus far onto the drone. This would involve getting the drone to autonomously fly and land on a stationary target. The next aim would be to autonomously track a moving target at a safe distance such as the roof of a car, before tracking it by hovering above the moving surface prior to landing on this object. In order to achieve this the classifier would need to be retrained for the target of interest. Additional sensors would also be added to the drone such as LiDAR or RADAR sensors might also be deployed to get a more accurate measurement data and increase the safety aspects of the drone design. This would improve the system design and combining a sensory entity with the computer vision would ensure the system handles real world unforeseen situations. This would also allow for a higher level of confidence in the drone's location.

During this time the robustness and reliability of the design could be further developed.

6.2.1 Stretch Goals

This project has a very broad scope and a limited time, If further time was provided it would initially be spent identifying a target in the vertical plane and testing methods to land the drone in a vertical plane. This would then be extended to tracking a moving target in the vertical plane and landing on it. For this to be achieved many aspects need to be considered such as the perching method of the drone, how it will attach and safely detach to the moving surface. This would be extremely valuable as it could then be used to optimise the structural health monitoring of wind turbines. The drone will then also have the capability to collect aerial imagery from wind turbines and dynamically measure and adduce wake pressure effects from the airflow surrounding the wind turbine.

6.3 Reflection

Looking back over the progression of the project, due to the broad scope of this project and the variety of challenges, there was no extended period of time when progress was stalled. If progress was impeded in one avenue, efforts could be refocused to another avenue for further exploration. However there are some things which would not be repeated if starting the project again.

For initial testing of the designed CNN model, a labelled dataset would have been used to test the performance of the architecture. Annotating a dataset is time-consuming work and while it did enable testing through a live video stream, a lot of time was invested in labelling targets and noise which could have been invested in carrying out other tasks.

A lot of time was also spent exploring different target recognition implementations. The `Tesseract` library was difficult to use and did not produce results as expected when used to read the identifiers on each target. Had a CNN model been trained with the labelled data originally, this would have been a better use of time.

Many target identification mechanisms were investigated that were not used in the final CNN model. However these all contributed to the final design, as with each implementation more knowledge was gained about the capability of `OpenCv` which eventually resulted in an accurate robust design.

6.4 Summary

This research has shown that there is real potential to improve the structural health monitoring systems of wind turbines by deploying drones and machine learning for the inspection. It has been demonstrated that moving targets can be identified and tracked through a machine learning system. Autonomous control was established in a simplified environment. While the combined implementation was not deployed on a drone for further testing , each component has been individually tested and evaluated yielding very promising results for the use of drones for active device inspection in complex contexts.

Bibliography

[1] R. Powlesland. Impacts of wind farms on birds: A review. *Report by New Zealand Deparment of Conservation*, (289):53, 2009.

[2] Nawal Cheggaga and Fatiha Ettoumi. A neural network solution for extrapolation of wind speeds at heights ranging for improving the estimation of wind producible. *Wind Engineering*, 35:33–54, 02 2011.

[3] Vibration control of wind turbine: recent advances and emerging trends. *International Journal of Sustainable Materials and Structural Systems*, 4:347, 2020.

[4] The irish wind energy association. `http://www.iwea.ie/index.cfm/page/technicalfaqs?#q28`. Accessed:23/03/2022.

[5] Wind Energy sustainable energy authority of ireland. `https://www.seai.ie/technologies/wind-energy/#:~:text=Wind%20energy%20in%20Ireland,of%20our%20total%20electricity%20demand`. Accessed:23/03/2022.

[6] Sandra Eriksson, Hans Bernhoff, and Mats Leijon. Evaluation of different turbine concepts for wind power. *Renewable and Sustainable Energy Reviews*, 12(5):1419–1434, 2008. ISSN 1364-0321.

[7] vetrotex. `https://www.vetrotextextiles.com/applications/wind-energy`. Accessed: 27/02/22.

[8] Anindya Ghoshal, Mannur J. Sundaresan, Mark J. Schulz, and P. Frank Pai. Structural health monitoring techniques for wind turbine blades. *Journal of Wind Engineering and Industrial Aerodynamics*, 85(3):309–324, 2000. ISSN 0167-6105.

[9] Hadi Sanati, David Wood, and Qiao Sun. Condition monitoring of wind turbine blades using active and passive thermography. *Applied Sciences*, 8(10), 2018.

[10] Office of energy efficiency renewable energy. `https://www.energy.gov/eere/articles/wind-turbines-bigger-better`. Accessed:23/03/2022.

[11] Breiffni Fitzgerald, Biswajit Basu, and S. Nielsen. Active tuned mass dampers for control of in-plane vibrations of wind turbine blades. *Structural Control and Health Monitoring*, 20, 2013.

[12] K. Thomsen, J. Petersen, E. Nim, Stig Øye, and B. Petersen. A method for determination of damping for edgewise blade vibrations. *Wind Energy*, 3:233 – 246, 2000.

[13] Tarek Berghoutu Mohamed Benbouzid and Xiandong Ma. Intelligent condition monitoring of wind power systems: State of the art review. *Energies*, 14(18):5967, 2021.

[14] Ashim Khadka, Arash Afshar, Mehrdad Zadeh, and Javad Baqersad. Strain monitoring of wind turbines using a semi-autonomous drone. *Wind Engineering*, 2021.

[15] Douglas Adams, Jonathan White, Mark Rumsey, and Charles Farrar. Structural health monitoring of wind turbines: method and application to a hawt. *Wind Energy*, 14(4): 603–623, 2011.

[16] Chia Chen Ciang, Jung-Ryul Lee, and Hyung-Joon Bang. Structural health monitoring for a wind turbine system: a review of damage detection methods. *Measurement Science and Technology*, 19(12):122001, 2008.

[17] Ruizhen Yang, Yunze He, and Hong Zhang. Progress and trends in nondestructive testing and evaluation for wind turbine composite blade. *Renewable and Sustainable Energy Reviews*, 60:1225–1250, 2016. ISSN 1364-0321.

[18] McGugan M Borum K K and Broendsted P. Condition monitoring of wind turbine blades, 2006.

[19] Lina Draudvilienė, Asta Meškuotienė, Renaldas Raišutis, Paulius Griškevičius, Žaneta Stasiškienė, and Egidijus Žukauskas. The usefulness and limitations of ultrasonic lamb waves in preventing the failure of the wind turbine blades. *Applied Sciences*, 12(4), 2022.

[20] Byeongjin Park, Hoon Sohn, Chul-Min Yeum, and Thanh C Truong. Laser ultrasonic imaging and damage detection for a rotating structure. *Structural Health Monitoring*, 12(5-6):494–506, 2013.

[21] Abhishek Reddy, V. Indragandhi, Logesh Ravi, and V. Subramaniyaswamy. Detection of cracks and damage in wind turbine blades using artificial intelligence-based image analytics. *Measurement*, 147:106823, 2019.

[22] C Eschmann, C.-M Kuo, C.-H Kuo, and Christian Boller. Unmanned aircraft systems for remote building inspection and monitoring. 01 2012.

[23] Dongwon Jung and Panagiotis Tsiotras. Modeling and hardware-in-the-loop simulation for a small unmanned aerial vehicle. page 2768, 2007.

[24] Marco La Civita, George Papageorgiou, William Messner, and Takeo Kanade. Design and flight testing of a high-bandwidth h-infinity loop shaping controller for a robotic helicopter. In *AIAA Guidance, Navigation, and Control Conference and Exhibit*, page 4836, 2002.

[25] Belkacem Kada and Y Ghazzawi. Robust pid controller design for an uav flight control system. In *Proceedings of the World congress on Engineering and Computer Science*, volume 2, pages 1–6, 2011.

[26] David Mader, R. Blaskow, Patrick Westfeld, and C. Weller. Potential of uav-based laser scanner and multispectral camera data in building inspection. *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, XLI-B1:1135–1142, 06 2016.

[27] Dario Floreano and Robert J Wood. Science, technology and the future of small autonomous drones. *nature*, 521(7553):460–466, 2015.

[28] Shuaijun Wang, Fan Jiang, Bin Zhang, Rui Ma, and Qi Hao. Development of uav-based target tracking and recognition systems. *IEEE Transactions on Intelligent Transportation Systems*, 21(8):3409–3422, 2020.

[29] Matthias Mueller, Neil Smith, editor="Leibe Bastian Ghanem, Bernard", Jiri Matas, Nicu Sebe, and Max Welling. A benchmark and simulator for uav tracking. In *Computer Vision – ECCV 2016*, pages 445–461. Springer International Publishing, 2016.

[30] Martin Danelljan, Goutam Bhat, Fahad Shahbaz Khan, and Michael Felsberg. Eco: Efficient convolution operators for tracking. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[31] Aisha Ajmal, Christopher Hollitt, Marcus Frean, and Harith Al-Sahaf. A comparison of rgb and hsv colour spaces for visual attention models. In *2018 International Conference on Image and Vision Computing New Zealand (IVCNZ)*, pages 1–6, 2018.

[32] Sam Hare, Stuart Golodetz, Amir Saffari, Vibhav Vineet, Ming-Ming Cheng, Stephen L. Hicks, and Philip H. S. Torr. Struck: Structured output tracking with kernels. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38(10):2096–2109, 2016.

[33] Christoph Feichtenhofer, Axel Pinz, and Richard P. Wildes. Spatiotemporal residual networks for video action recognition. *CoRR*, abs/1611.02155, 2016.

[34] Joseph Redmon and Ali Farhadi. Yolo9000: Better, faster, stronger. pages 6517–6525, 07 2017.

[35] Jin Kim, Yoon Suk Lee, Sang Su Han, Sang Ho Kim, Gun Hyuk Lee, Ho Jun Ji, Hye Ji Choi, and Kwang Nam Choi. Autonomous flight system using marker recognition on drone. In *2015 21st Korea-Japan Joint Workshop on Frontiers of Computer Vision (FCV)*, pages 1–4. IEEE, 2015.

[36] M. Anwar Ma'sum, M. Kholid Arrofi, Grafika Jati, Futuhal Arifin, M. Nanda Kurniawan, Petrus Mursanto, and Wisnu Jatmiko. Simulation of intelligent unmanned aerial vehicle (uav) for military surveillance. pages 161–166, 2013.

[37] Xinyun Chen, Chang Liu, Bo Li, Kimberly Lu, and Dawn Song. Targeted backdoor attacks on deep learning systems using data poisoning. *CoRR*, 2017.

[38] Tom B. Brown, Dandelion Mané, Aurko Roy, Martín Abadi, and Justin Gilmer. Adversarial patch. *CoRR*, abs/1712.09665, 2017.

[39] Yellow car clip art. `https://www.clipartmax.com/middle/m2i8i8K9d3K9m2Z5_clipart-car-birds-eye-view-bird-s-clip-art-library-birds-eye/`. Accessed: 01/04/22.

[40] Wind turbine blade image. `https://windenergy.dtu.dk/english/news/2018/09/long-wind-turbine-blades-require-new-composite-materials?id=1bafd1fa-4578-4ab9-b212-d01a5e204260`. Accessed on:01/04/2022.

A1 Appendix

Abbreviations

AE	Acoustic Emission
AI	Artificial Intelligence
ATMD	Active Tuned Mass Dampers
CNN	Convolutional Neural Network
FBG	Fiber Bragg Grating strain sensors
FOV	Field of View
GBR	Ground Based Radar
HAWT	Horizontal Axis Wind Turbine
HSV	Hue Saturation Value
ML	Machine Learning
NDT	Non-Destructive Testing
NMS	Non Maximum Suppression
ODS	Operational Deflection Shapes
RC	Resonant Comparison
RGB	Red Green Blue
ROI	Region of Interest
SEAI	Sustainable Energy Authority of Ireland
SHM	Structural Health Monitoring
SS	Selective Search
TF	Transmittance Functions
UAV	Unmanned Aerial Vechicle
UT	Ultrasonic Testing
VAWT	Vertical Axis Wind Turbine
VTOL	Vertical Take-Off and Landing
WP	Wave Propagation
WT	Wind Turbine
WTB	Wind Turbine Blade
YOLO	You Only Look Once