# Third-Person Autonomous Control using Deep RC

Jaron Ellingson[1], Kameron Eves[2], Nick Walton[3], and Cameron Peterson[4]

***Abstract*— The ability to fly a remote control (RC) aircraft from a third-person perspective is a skill that many hobbyists and enthusiasts enjoy. With a little practice, an RC pilot can sense the aircraft's orientation and apply the correct inputs for it to orbit, achieve specific mission objectives, or line-of-sight waypoints.**

**The work done in [1] proved that third-person sensing of an aircraft's attitude is possible. This work seeks to improve the deep learning methods, increase hardware capabilities through the addition of a turret and motion capture validation, and add control algorithms for complete autonomous flight. While the results are not comprehensive, this report gives the work accomplished thus far. Overall this report will show that substantial progress has been made towards third-person autonomous control.**

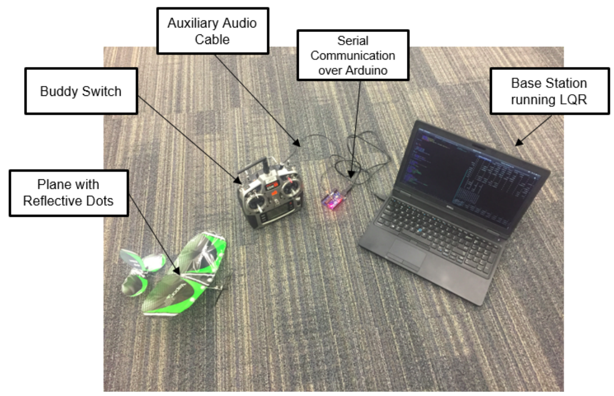



Fig. 1: This paper will demonstrate a feasible method for estimating and controlling the state of an RC aircraft using deep learning and only third-person-perspective visual sensing. This figure shows the Vapor Lite by Horizon Hobby which was selected for testing because of it's ability to fly slow with in a motion capture room. The other objects are useful in running the control algorithms. The RC transmitter is programed with a buddy box system which allows the pilot to easily switch whether the base station algorithm or the pilot is controlling the aircraft.

## I. Introduction

Unmanned aircraft systems (UAS) have become a staple of modern life. From package delivery to military airstrikes, from law enforcement surveillance to film videography, from search and rescue to hobbyist out on the weekend, drones have become ubiquitous with modern progress and have expanded human capability. While UAS can be piloted via remote control signal, the ability to operate autonomously does expand the usefulness of UAS exponentially by offering improved performance and reduced operator workload. If unmanned aircraft are to be used for applications such as the prescient air taxi, they must be sufficiently autonomous for the lay person to use.

### A. Background (UAS)

Advances in autonomous UAS technology have primarily focused in four areas: physical systems, control algorithms, sensing modalities, and applications. Each of these four research thrusts are essential to modern UAS technology. Because this work seeks to bring together each of these thrust into one system as well as improve upon one specific thrust (sensing modalities), a brief introduction to and survey of each area follows.

**Physical Systems**. Today, technology has progressed to the point to allow for autonomous systems on helicopters and quadcopters to excel. Significant progress has been made in the field of aerodynamics allowing for increased efficiency. Of course, physical systems include more than the airframes and platforms. Advancements in electrical technology (particularly the ratio between computational power vs physical weight) have been extremely important. While radio transmitters and electronic speed controllers have applications in all UAS flight, electrical advancements have been particularly important in autonomous UAS.

This work adds little to physical systems research. However, this work does heavily use these advances and would not be possible without the countless hours that have made modern UAS physical systems possible

**Control Algorithms**. Since proportional-integral-derivative (PID) control was formalized by Nicolas Minorsky in 1922, it has been far and away the most well-known and well used system governor. Today, PID control is still the most common control algorithm, however, modern control theory points towards full-state-feedback (FSF) as a potential usurper to the omnipresence of PID. This paper demonstrates the effectiveness of FSF in aerospace applications, by applying a FSF controller to a fixed wing UAS. This is done through a linear-quadratic-regulator (LQR) which optimizes the gains used in the FSF controller.

**Sensing Modalities**. To autonomously control UAS, it is essential for the aircraft to receive measurements updating its state. This field of research includes everything from

[1]Jaron Ellingson is a graduate student in the Department of Mechanical Engineering, Brigham Young University jaronce@byu.edu

[2]Kameron Eves is an undergraduate in the Department of Mechanical Engineering, Brigham Young University ccackam@byu.edu

[3]Kameron Eves is an undergraduate in the Department of Computer Science, Brigham Young University nickwalton@byu.edu

[4]Cameron Peterson is a professor in the Department of Electrical Engineering, Brigham Young University cammy.peterson@byu.edu

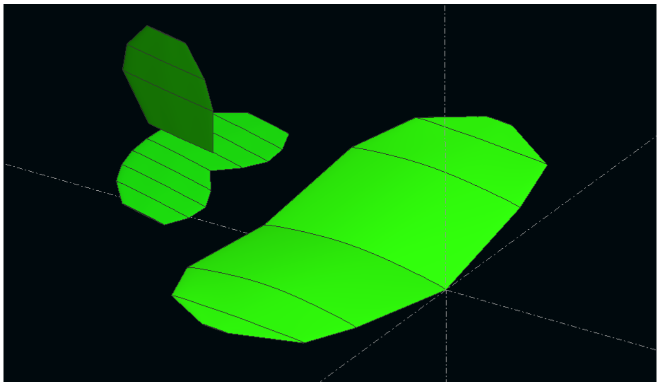


Fig. 2: Screenshot from the XFLR5 program. A user can define the airfoil shapes and overall structure of the plane. Here the Vapor Lite is modeled. The program runs through computational fluid dynamic equations to obtain the aerodynamic coefficients of the aircraft which can be used in the overall dynamic model identification.

accelerometers, rate gyros, and laser range finders to Extended Kalman Filters (EKF) and simultaneous localization and mapping (SLAM). This work seeks to add much to this field by introducing a novel form of sensing and proving its effectiveness.

**Applications**. This research thrust is by far the most diverse and includes studies such as motion planning, mission planning, and target tracking. While this work does not directly contribute this field of research, it does open avenues for further research in this field by providing a novel approach to sensing and controlling the aircraft.

### *B. Background (Deep Learning)*

Deep Learning is an exciting area of machine learning that has rapidly advanced in the last several years. It has improved the state of the art for computer vision, machine translation, reinforcement learning and many other fields. Unmanned aerial systems have specifically benefited from deep learning. [2] summarizes many different methods and applications including indoor/outdoor navigation, feature extraction, scene classification, path planning, and motion control. These methods operate in supervised, unsupervised, and reinforcement learning domains and assume sensing technology on the aircraft, such as image, acoustic, radar, and lidar. While these approach the sensing and estimation from onboard the aircraft, this work provides a novel third-person sensing modality.

### *C. Deep RC*

As mentioned above, this work seeks to further modern UAS research by utilizing a new sensing modality. This sensing modality is a third person state estimator capable of determining an aircrafts orientation via a monocular ground-based camera. This method was first introduced by [1]. This system functions by taking an image of the aircraft in flight, using machine learning to identify the aircrafts orientation, and propagating the aircraft's state using that orientation and a particle filter. In [1] simulation results demonstrate that the estimated state converge to the true state. However, they were not able to completely demonstrate the system in hardware. While hardware results did show some correct state estimation, the cameras field of view, the cameras resolution, and lighting differences between training and testing data all inhibited the algorithms performance.

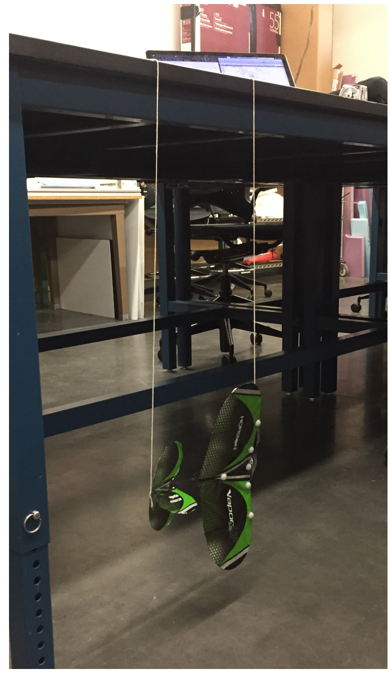


Fig. 3: Photo of capturing the mass moment of inertia. The aircraft is was suspended from the table and then swung around its center to get a final inertia result.

This work is a continuation of the work in [1]. It seeks to improve the work by providing a comparative study of deep learning approaches, a more robust tracking system, and a complete third-person control. This paper highlights the work accomplished and seeks to prove that full autonomous control is achievable.

This work is similar to [3] which also applies machine learning to estimate state through a camera. However, that research involves a camera mounted to the underside of the UAS as well as prior knowledge of the location, size, and orientation of ground targets. [4] also demonstrated end to end control of a UAS with a camera and machine vision. However, their work focused primarily on navigation (the applications category) and has relatively little application to state estimation.

These similar studies all have one commonality. That is, they require the camera to be mounted on the aircraft itself. To the authors knowledge this work and the work it builds on by [1] is the only method and/or demonstration of state estimation via third person (i.e. ground based) camera.

The advantages and applications of such a system are numerous. Advantages include removing accelerometers, rate gyros, barometers, global positioning systems (GPS), and magnetometers. This can provide significant weight reduc-

Fig. 4: The result from tracking the Vapor Lite in the motion capture room. Notice that the turret has motion capture dots as well as the Vapor Lite. Their respective orientations are known from the motion capture data and the needed desired pan and tilt for the turret is calculated.

tions. Weight is a major factor in all flight, and so any reduction can provide significant improvements in UAS performance (i.e. endurance, range, stall speed, maneuverability, stability, etc). The potential for redundant systems should not be ignored as well.

Applications of this state estimation method are also numerous. Take for example a noncooperative or even adversarial aircraft that is impinging upon restricted air space. Provided some method of arresting control of this aircraft from its operator, a method such as this could provide state estimation for a control algorithm to maneuver this aircraft to and land it at a safe place. Finally, pit falls such as the necessity of line of sight make loss of GPS a common problem. Operation in GPS denied environments is another great application of this technology. Search and rescue missions indoors or in canyons could benefit greatly from this technology. This system can be thought of as a portable and flexible (if slightly less accurate) motion capture system which requires prior knowledge of the systems dynamics.

More abstractly, this research adds to the body of knowledge regarding the capabilities of machine learning. This system is bio-inspired and replicates the human RC pilots process of visually observing an aircraft in flight and estimating its state purely from these visual observations. Demonstrating this system allows for comparison between artificial neural networks and biological neurology.

## II. Methodology

To accomplish the above goals, an end-to-end autonomous UAS which utilizes this deep RC technique was developed. While this system is similar in function to the previous works system there were several important changes which made a hardware realization of this system possible. To overcome the previous works limitations, the following actions were taken:

- To remove the effect of variations in lighting, the system was developed indoors using a very small light weight aircraft that could operate in confined quarters.
- To remove the difficulty of the aircraft leaving the cameras field of view a gimbal control was implemented. This allowed the camera to track the aircraft in flight (Fig 4).
- To improve the quality of the images, a camera specifically built for machine vision was identified and chosen.
- To improve the validation accuracy and iteration time it takes to validate on hardware, a system was developed to fly an RC aircraft in the motion capture room (Fig 1)

In this section we describe the equipment used, the methods for obtaining true orientation state of the aircraft, the tracking algorithm and hardware, and finally describe the type of control algorithm used.

### A. Equipment

The Blackfly S model, BFS-U3-200S6, camera made by FLIR integrated Imaging Systems Inc. was used with a Computar V0828-MPY lens with an 8 mm focal length. This assembly was mounted on a PhantomX Turret Kit sold by Trossen Robotics. This gimble allowed for both pan and tilt camera motion. A UMX Vapor Lite HP BNF Basic airframe sold by Horizon Hobby was used. An OptiTrack motion capture system was used to find the true state. This true state was used for comparison and neural net training purposes only.

In addition, a system to easily transmit commands to a airplane was created (Fig 1). This system utilizes the ability of a transmitter to perform what is called a buddy box pairing with the computer. The transmitter is connected to the computer via an auxiliary cable and arduino microcontroller which in turn is connected to the ROS network running on the base station. The arduino sends serial commands to the transmitter and allows the pilot to manually switch between the base station output from the control algorithm and from the pilot's outputs. This system allows for the pilot to recover the airplane if the control algorithm is not working and provides for faster iterations in tuning the control system.

### B. True States

Obtaining the true orientation is essential for evaluating the Deep RC method of state estimation and for training the neural net. The true orientation as seen from the camera was found by rotating the aircrafts rotation matrix from the vehicle frame to the cameras body frame. This system of equations was then solved for the aircrafts orientation.

### C. Tracking

The neural net used for determining the state of the aircraft also returned the pixel location of aircraft. The following equation was used to track the aircraft using this pixel location.

$$\Delta\psi = \arctan\left(\frac{P_{ia}}{P_{lr}}\tan\left(\frac{\phi}{2}\right)\right) \tag{1}$$

In this equation, $P_{ia}$ is the distance in pixels from the center of the image to the aircraft and $\Delta\psi$ is the angle change that must be sent to the gimbal. $P_{lr}$ and $\phi$ are respectively the image dimension in pixels and the cameras viewing angle and are constant parameters of the camera.

### D. LQR

The first step to control our system with a LQR controller was to create a dynamic model of the system. The full nonlinear dynamic model defined in [5] was used. The parameters necessary for this model were found using a combination of physical measurement (i.e. weight and scale) and XFLR5 aircraft design software which determines the aircrafts aerodynamic and stability parameters. Figure 2 shows the model of the Vapor Lite created and highlights how a user can define airfoil shapes for the wings as well as the horizontal and vertical stabilizers.

Finally, Figure 3 shows how we obtained the mass moments of interia. Suspending the aircraft from the table, we gave the airplane an impulse rotation and then counted the frequency of oscillations and calculated the time period of oscillations. Equation 2 gives the equation we used for finding the inertia where $g$ is gravity, $m_0$ is the mass of the object, $r^2$ is the radius or half the distance between the two strings, $\tau$ is the time period of one oscillation, and finally $s$ is the length of the cables.

$$J = \frac{g m_0 r^2 \tau^2}{4 s \pi^2} \tag{2}$$

## III. Deep Learning Approach

### A. Training Data

Data for initial network training and testing was gathered using Holodeck [6], a robotics simulator built using Unreal Engine. 10,000 images of the plane rotated at random angles and different backgrounds were gathered to use in training the networks. 2,000 were generated as a simulation test set. For each image with a background, an image of a plane at the same orientation without a solid gray background was also generated.

### B. Network Plane Tracking

To track the plane a pre-trained Single Shot Detection (SSD) network was used [7]. The network was trained on the COCO dataset which included a plane classification class which we were able to use to detect the position and bounding box of a plane in an image. With the bounding box we are able to crop the image to feed into the orientation estimation network.

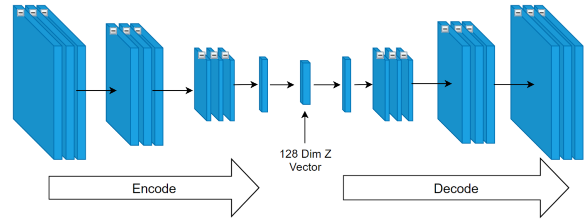
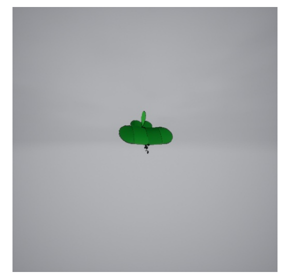



Fig. 5: A representation of our auto encoder architecture. The image is fed through several convolutional layers and then encoded into a 128 Dimensional vector by a fully connected layer. It is then decoded by using another fully connected layer followed by transposed convolutional layers.

(a) training input image



(b) training output image

Fig. 6: The auto encoder was trained to take as input images with background and output images with a blank background.

### C. Network Comparisons

We tested several different networks and compared their accuracy to determine the best model to use for initial orientation estimation: A ”Codebook” method and a direct regression network. Each method was tested for both single estimation accuracy and mixture of Gaussians accuracy.

**Codebook**. The Codebook method as outlined by [8] trains an auto encoder to encode a image into a 128 dimensional Z vector, then to decode it back into a similar image that retains all orientation information. This forces the network to learn to encode the orientation information directly into the Z vector. After training, a ”codebook” is generated by taking images of orientations at evenly spaced angles, and then recording their Z vectors and orientations. During prediction new images are encoded into a Z vector which is compared with all Z vectors in the codebook. The closest Z vectors are found, and their corresponding orientations are used as the predicted single or mixture of Gaussians orientation of the image to predict. A representation of this architecture can be seen in figure 5.

To encourage the network to only learn to save orientation information the auto encoder was trained to take as input images with some background and output images with a blank background. This encourages the model to leave out irrelevant background information when encoding the image and only store information about the plane and it's orienta-

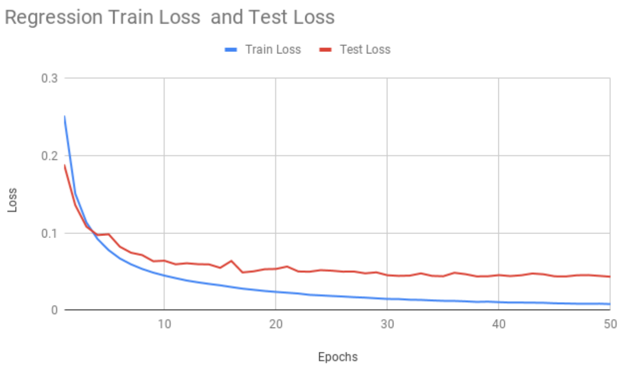



Fig. 7: Regression train and test loss on simulated data

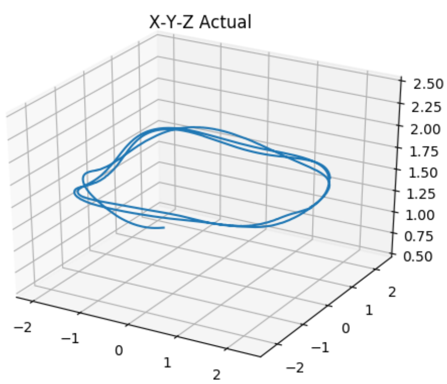



Fig. 8: The flight path of the Vapor Lite in the motion capture room.

tion. An example training pair can be seen in figure 6

**Regression**. The direct regression network was a standard feed forward CNN that attempted to directly regress the rotation matrix representation of the orientation. It contained four convolutional layers followed by two fully connected layer.

## IV. Results

### A. Hardware

The improvements to the hardware can be seen in Figures 1 and 4. The resulting images are a $5472x3648$ image which is then cropped down to a $128x128$ size image. The results of the cropped images can be seen in Figure 12 on column 1.

### B. Classification

The results for both the codebook and regression networks are given in Table I. These results show the error for roll, pitch, yaw, and overall average in degrees. The results from both networks will be discussed in this section.

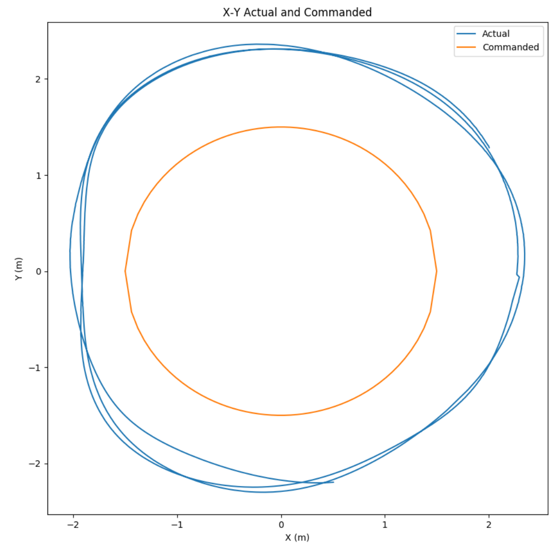



Fig. 9: Actual and commanded flight path of the Vapor Lite in the motion capture room on the x-y plane. The radius of the flight orbit of the aircraft was roughly one-half meter error.

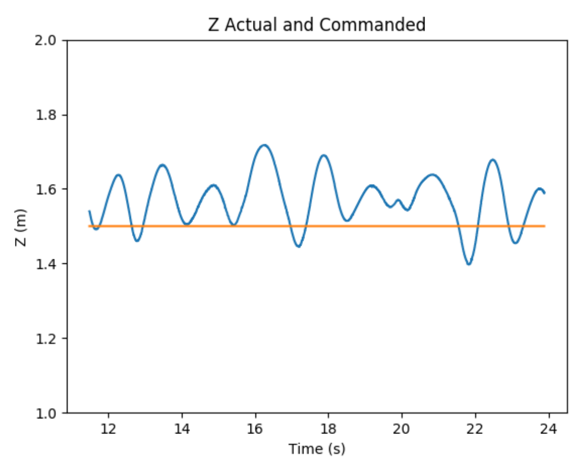



Fig. 10: Actual and commanded flight altitude of the Vapor Lite in the motion capture room. The altitude of the aircraft was roughly one-fifth meter error.

TABLE I: Network Classification Result Error (degrees)

| | | Roll | Pitch | Yaw | Average |
|---|---|---|---|---|---|
| Codebook | Simulation | 21.38 | 4.79 | 22.86 | 16.34 |
| | Real | 87.43 | 21.77 | 80.18 | 63.17 |
| Regression | Simulation | – | – | – | 0.15 |
| | Real | 159.62 | 14.97 | 83.43 | 86.01 |

**Codebook**. The output images from the codebook network are shown in Figure 11, where the top five most orientations are given. The first column shows that input images and the rest of the columns show the outputs with the most similar output being farthest to the left. The simulation results show that the codebook can accurately, at least within human level perception, identify the orientation of the aircraft. Rows 3 and 4 show that the network is not immune to the bimodal nature of classifying aircraft with a single image. As discussed in [1], RC pilots can also get confused about the orientation but have to "trust the sticks" to remain on course. For this reason, the top 5 outputs of the network are taken into consideration instead of the top single best output.

Figure 12 comparatively shows the output of the network on real data. While row 1 shows what could be fairly accurate results, rows 2-4 show significantly less accurate than the simulated data. These images correspond with the high error report in Table I. We believe this discrepancy in accuracy is largely due to the low resolution of the testing images as well as the significant differences between the training data and the test data. The training data used a rough model of the plane, with only a solid green wing pattern as opposed to the mixed black and green pattern of the actual plane. The plane model was also perfectly rigid, unlike the real world plane. These plane model differences as well as the reduced resolution of the test data likely prevented the network from being able to generalize to the significantly different test dataset.

These failures however are relatively easy to solve. The low resolution images can be improved with a different camera or perhaps with zoom lens. The model inaccuracy could be remedied by using real world data as well as simulated data to train the model.

**Regression**.

The regression network had similar results as the codebook network in that it seems to learn really well when training on simulation data. Figure 7 shows that the network begins to learn the correct orientations and Table I concurs. After 50 epochs the final average error for any orientation was roughly 0.15 degrees but when testing on real data the average error jumps to 86.01 degrees.

### *C. Autonomous Control*

The results from the LQR controller are shown in Figures 8 to 10. Figure 8 shows the complete flight path of the vehicle while Figures 9 and 10 show the commanded and actual flight paths of the vehicle. These results show that autonomous flight is possible and is a good starting point for more accurate control. It is expected that with a little more tuning, the commanded orbit and altitude will be achieved.

## V. Conclusions and Future Work

Our major accomplishments in this research include turret tracking an aircraft, LQR control, and promising deep learning results. Additionally, we were able to create a way to easily test in a motion capture room once the complete system is ready.

The research reported above is missing the link between the deep learning classification and the actual control of the aircraft. While this report shows great progress, the research will need to revisit the particle or kalman filter approaches to convert the classification estimates into states of the aircraft. Extensive testing needs to be conducted both in simulation and hardware. The motion capture room will once again be beneficial in comparing truth to estimated states.

Another future area of work is improving the deep learning classification and comparing it specifically to the shapenet classification in [1]. Training each network on the exact same data and extensive testing on real data will ensure that the optimal network is used.

A major key to success will be creating a high fidelity simulation. The Holodeck simulation software provides an efficient method for improving our methods because it allows for fast iterating on our algorithms while maintaining a visually accurate environment to test in.

## References


[1] Jaron Ellingson, Gary Ellingson, and Tim McLain. Deep rc: Enabling remote control through deep learning. 2018.

[2] Dhiraj Gandhi, Lerrel Pinto, and Abhinav Gupta. Learning to fly by crashing. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3948–3955. IEEE, 2017.

[3] M Mérida-Floriano, F Caballero, D García-Morales, F Casares, and L Merino. Bioinspired vision-only uav attitude rate estimation using machine learning. In *2017 International Conference on Unmanned Aircraft Systems (ICUAS)*, pages 1476–1482. IEEE, 2017.

[4] Benjamin S Chiel and John Baillieul. Visual gps-denied multi-agent localization & terrain classification. In *2018 IEEE Aerospace Conference*, pages 1–12. IEEE, 2018.

[5] Randal W Beard and Timothy W McLain. *Small unmanned aircraft: Theory and practice*. Princeton university press, 2012.

[6] Joshua Greaves, Max Robinson, Nick Walton, Mitchell Mortensen, Robert Pottorff, Connor Christopherson, Derek Hancock, and David Wingate. Holodeck: A high fidelity simulator, 2018.

[7] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. Ssd: Single shot multibox detector. In *European conference on computer vision*, pages 21–37. Springer, 2016.

[8] Martin Sundermeyer, Zoltan-Csaba Marton, Maximilian Durner, Manuel Brucker, and Rudolph Triebel. Implicit 3d orientation learning for 6d object detection from rgb images. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 699–715, 2018.

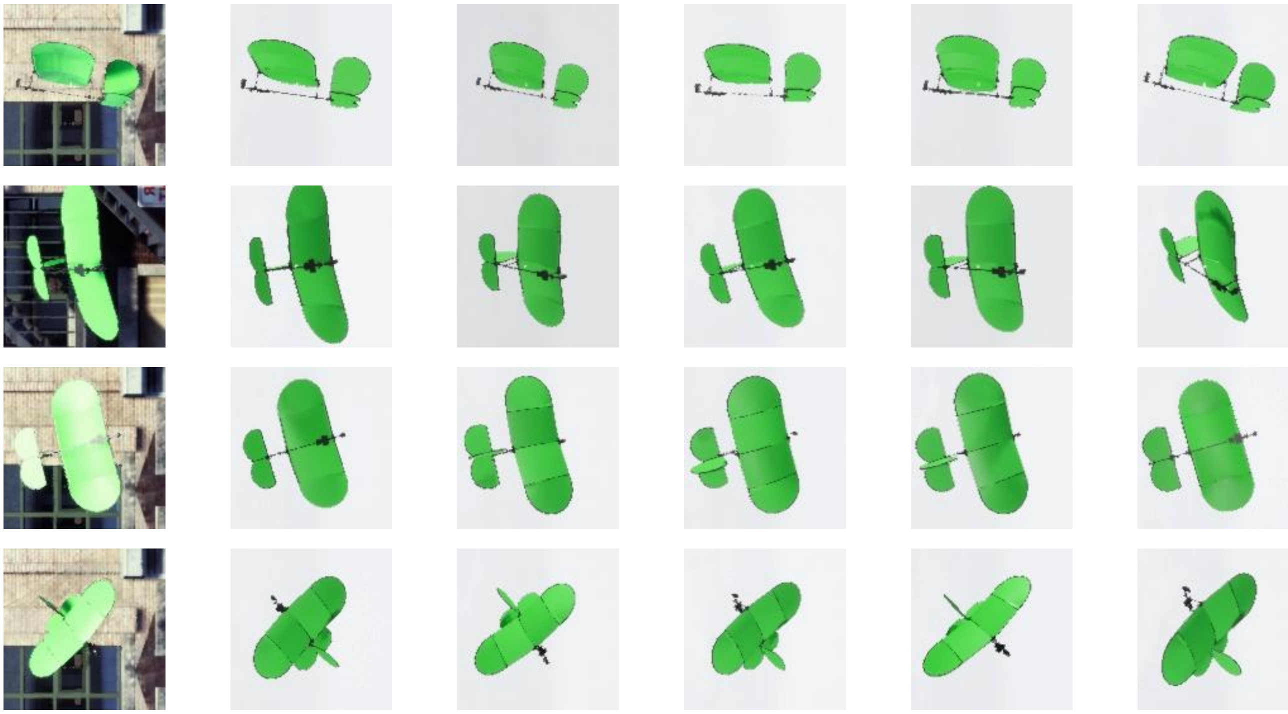


Fig. 11: Example simulation testing images and the output from the codebook network. The first column is the testing image and the other columns are the first 5 images from the code book which correspond to the classification. Notice how each of the images are close to the original. In addition, rows 3 and 4 show the bimodal nature of the classifying the aircraft's orientation. In particular row 4 guesses the wrong, but flipped 180 degrees, orientation on image 1, 3, and 5 but guesses the correct orientation on image 2 and 4.

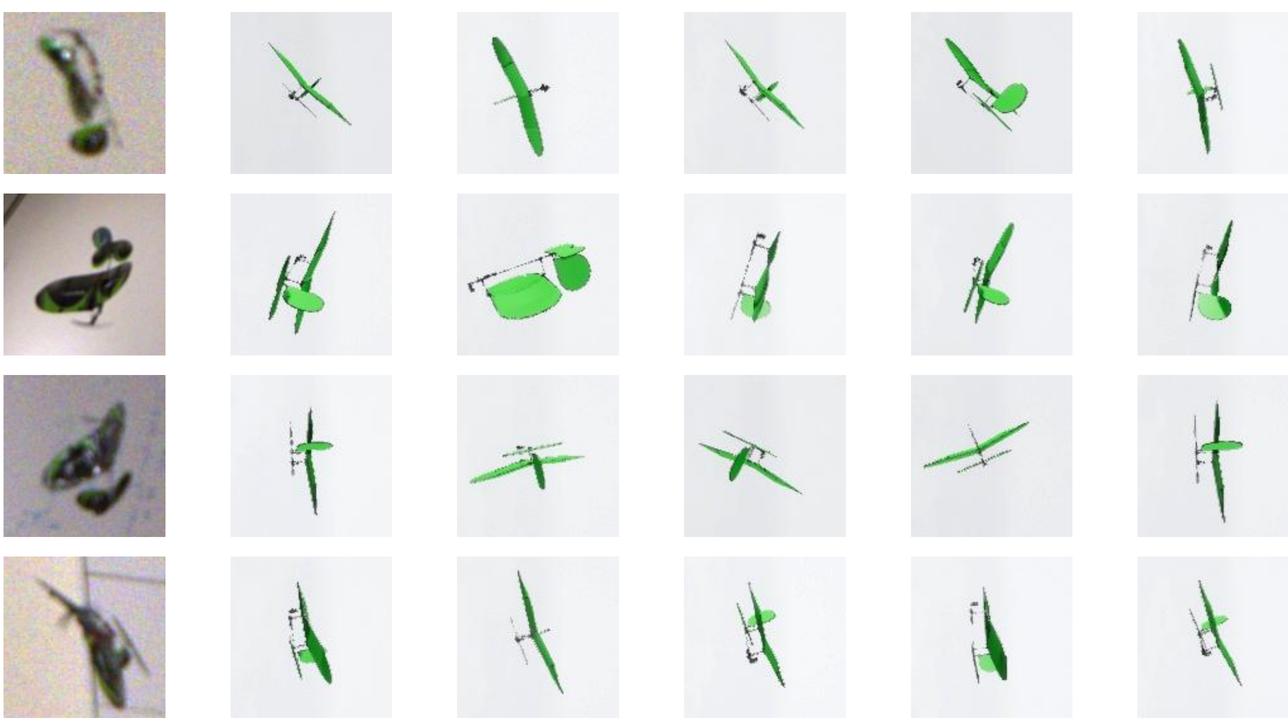


Fig. 12: Example hardware testing images and the output from the codebook network. Similar as the above image, the first image is the testing image and the remaining images in a row are the images which are selected from the codebook network which most accurately align with the test image. These results are not as accurate as the simulation results due to a lower resolution camera and inaccurate simulation model.