



All Student Publications

2018-09-19

Deep RC: Enabling Remote Control through Deep Learning

Jaron Ellingson

Brigham Young University, jaronce@byu.edu

Gary Ellingson

Brigham Young University, gary.ellingson@byu.edu

Tim McLain

Brigham Young University, mclain@byu.edu

Follow this and additional works at: <https://scholarsarchive.byu.edu/studentpub>



Part of the [Mechanical Engineering Commons](#), and the [Robotics Commons](#)

BYU ScholarsArchive Citation

Ellingson, Jaron; Ellingson, Gary; and McLain, Tim, "Deep RC: Enabling Remote Control through Deep Learning" (2018). *All Student Publications*. 241.

<https://scholarsarchive.byu.edu/studentpub/241>

This Peer-Reviewed Article is brought to you for free and open access by BYU ScholarsArchive. It has been accepted for inclusion in All Student Publications by an authorized administrator of BYU ScholarsArchive. For more information, please contact scholarsarchive@byu.edu, ellen_amatangelo@byu.edu.

Deep RC: Enabling Remote Control through Deep Learning

Jaron Ellingson¹, Gary Ellingson², and Tim McLain³

Abstract—Human remote-control (RC) pilots have the ability to perceive the position and orientation of an aircraft using only third-person-perspective visual sensing. While novice pilots often struggle when learning to control RC aircraft, they can sense the orientation of the aircraft with relative ease. In this paper, we hypothesize and demonstrate that deep learning methods can be used to mimic the human ability to perceive the orientation of an aircraft from monocular imagery.

This work uses a neural network to directly sense the aircraft attitude. The network is combined with more conventional image processing methods for visual tracking of the aircraft. The aircraft track and attitude measurements from the convolutional neural network (CNN) are combined in a particle filter that provides a complete state estimate of the aircraft. The network topology, training, and testing results are presented as well as filter development and results. The proposed method was tested in simulation and hardware flight demonstrations.

I. INTRODUCTION

Small unmanned aircraft systems (SUAS) have the potential to be revolutionary in multiple industries including: cinematography, agriculture, infrastructure monitoring, and even automatic package delivery. While significant advances have been made, our research aims to enhance SUAS capabilities by introducing a novel aircraft attitude estimation system.

This work focuses on the problem of remote or third-person sensing and estimation of a SUAS and is similar to an RC pilot standing far from a aircraft while maintaining the ability to fly it. While novice RC pilots may have limited ability to properly control the aircraft, they can instinctively detect the state of the aircraft using only remote, visual sensing. In particular, sensing the orientation of the aircraft is not difficult for RC pilots but is nearly impossible for traditional computer-vision algorithms, without additional aids such as fiducial markers.

In recent years, a new vision processing method has been developed using deep learning for additional sensing. Deep networks for visual sensing or convolutional neural networks (CNNs) have been demonstrated to have near or better than human performance in autonomous navigation and control tasks. An innovative example shown in [1] shows that CNNs

*This work is supported by the Brigham Young University Harold B Lee research grant and partially supported by the Center for Unmanned Aircraft Systems (C-UAS), a National Science Foundation/University Cooperative Research Center (I/UCRC) under NSF award No. IIP-1161036 along with significant contributions from C-UAS industry members.

¹Jaron Ellingson is beginning as a graduate student (work done as an undergraduate) in the Department of Mechanical Engineering, Brigham Young University jaronce@byu.edu

²Gary Ellingson is a PhD candidate in the Department of Mechanical Engineering, Brigham Young University gary.ellingson@byu.edu

³Tim McLain is a professor in the Department of Mechanical Engineering, Brigham Young University mclain@byu.edu



Fig. 1. This paper will demonstrate a feasible method for estimating the state of an RC aircraft, such as these, using deep learning and only third-person-perspective visual sensing. This UMX Cessna 182 and UMX Timber by Horizon Hobby were selected for testing the proposed method because they are visually representative of the high-wing airplanes used to train the CNN.

can learn a controller strategy that performs at an expert first-person pilot level.

Our research is similar, but instead of using CNN for end-to-end control of the aircraft, we approach the problem by focusing on only the vehicle state estimation that utilizes the CNN for orientation sensing from third-person-perspective monocular imagery. Using the CNN as a sensor to enable classical state estimation will also allow the filter to rely on the Markov assumption to incorporate previous measurements and aircraft states.

In this paper, we first present several pertinent and related works followed by the creation of an attitude sensing CNN, including our method for creating and augmenting the training data, training the network, and validation testing. We then describe how CNN measurements can be combined with more classical image processing and estimation techniques to observe the full state (position and orientation) of an RC aircraft. Finally, we present the results of simulation and hardware flight demonstrations.

II. RELATED WORKS

Deep learning techniques have recently shown exciting promise in a large variety of academic fields. CNNs have been demonstrated to have near or better than human performance in tasks ranging from machine vision [2]–[5] to playing the game of Go [6]. Other examples include face recog-

nition [7], speech recognition [8]–[11], cancer cell recognition [12], vehicle classification [13], and robotics [14].

The dataset for training a CNN to identify the orientation of objects was produced by researchers at Stanford and Princeton Universities [15]. Called ShapeNet, the work created a renderer for producing images from three dimensional graphic models. The models can be quickly reproduced in a specified orientations with random variations including lighting, skewing, and backgrounds.

Using a CNN classifier, the work in [16] demonstrated a method for localization and landing zone detection for a parafoil aircraft. The work in [17] used a CNN to help estimate the attitude of an aircraft using an on-board camera. These works both show a CNN being used as a sensor in Bayesian filters for state estimation.

The output of the CNN is computed through a softmax (or normalized exponential) function [18]. This function allows for a probability $P(\mu|x)$ to be computed given some input x . This probabilistic representation of the output fits nicely into state-of-the-art estimation and control theory, such as particle and Kalman filters which function is based on stochastic measurements and probabilistic state uncertainty.

Particle filters have been used in a variety of applications from multi-target robotic tracking [19] to nonlinear state estimation [20] and they are particularly well suited for robot-localization problems. The benefits of a particle filter are that the observation models need not be to linearized as required for the Kalman filter or extended Kalman filter, respectively, and they do not require the formulation of Jacobian matrices, which can be complicated or even impossible to compute in some situations [21].

III. BUILDING THE CNN

In constructing the CNN, several factors were carefully considered: creating realistic training data, training the network to produce attitude measurements, and validating the network on real imagery.

A. Dataset

To create a training dataset, all of the high-wing, piston-powered airplane models, such as the Piper J-3 Cub or the Cessna 172, were selected from the ShapeNet database. The ShapeNet model render interface was then used to produce a variety of specified orientations. At this stage, the output images had a transparent background and were created with random lighting and skewing.

In addition to the rendering, the airplane model images were placed at a semi-random location on a realistic background. This was done to ensure that the CNN was trained on data representative of real imagery and the CNN would learn to ignore the background and focus on the aircraft. Some of the training images have a blue sky background while others have a mountain or grassy background. These images were finally resized to a 50×50 resolution and were then used to train the complete CNN network. An example of the training data creation process is shown in Figure 2.

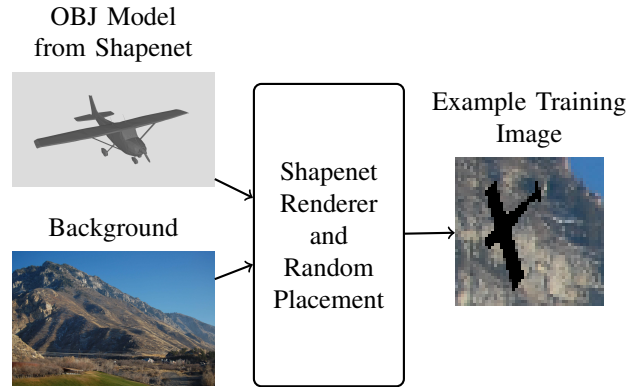


Fig. 2. An example of the training images. Each image contains a rendered model and a realistic background. The training images from ShapeNet’s renderer were reduced in size and slightly offset from the background image center. This provided the network with more realistic data. These images are representative of low resolution images of an RC aircraft, but despite the low resolution the orientation of the aircraft can still be perceived.

Overall, the data consisted of 23 different airplane models each with a 1000 random views generated. The data was further augmented by rotating these views around the center of the image, similar to the image augmentation described in [17]. The last complete set of model images and the last 50 images from each model were set aside as testing data to monitor overfitting. At every training iteration the prepared modeled images, with their corresponding truth angles, were fed to the network in batches of 50.

B. Training the Network

To train the network, several depths and learning weights were considered, but a seven-layer network was chosen to keep the training simple. This configuration enabled the network to be trained quickly with limited computation resources and provides a proof-of-concept network to show the viability of the system approach. The final architecture includes four convolutional layers with two fully-connected layers, followed by a final fully-connected output layer corresponding to the attitude-angle probability distributions. Every layer had a rectified linear unit (ReLU) for the activation function except for the final layer, which used three softmax probabilities (one for each attitude angle). An adaptive learning rate was used to aid training and a 0.5 dropout probability helped prevent overfitting.

The network’s loss function is similar to what was used in the original ShapeNet CNN [22]. The main difference is that, instead of calculating the probability of classifying the right class, we consider a only single model type: airplane. Additionally, a simple angular distance d was used instead of the geodesic distance between yaw and pitch. The loss L_θ was calculated from the function

$$L_\theta = - \sum_{\angle} \sum_{\nu \in V} e^{-d} \log P(\mu_\nu | x)$$

where the first sum is over the three aircraft orientation angles (ϕ , θ , and ψ) and the second sum is over V the

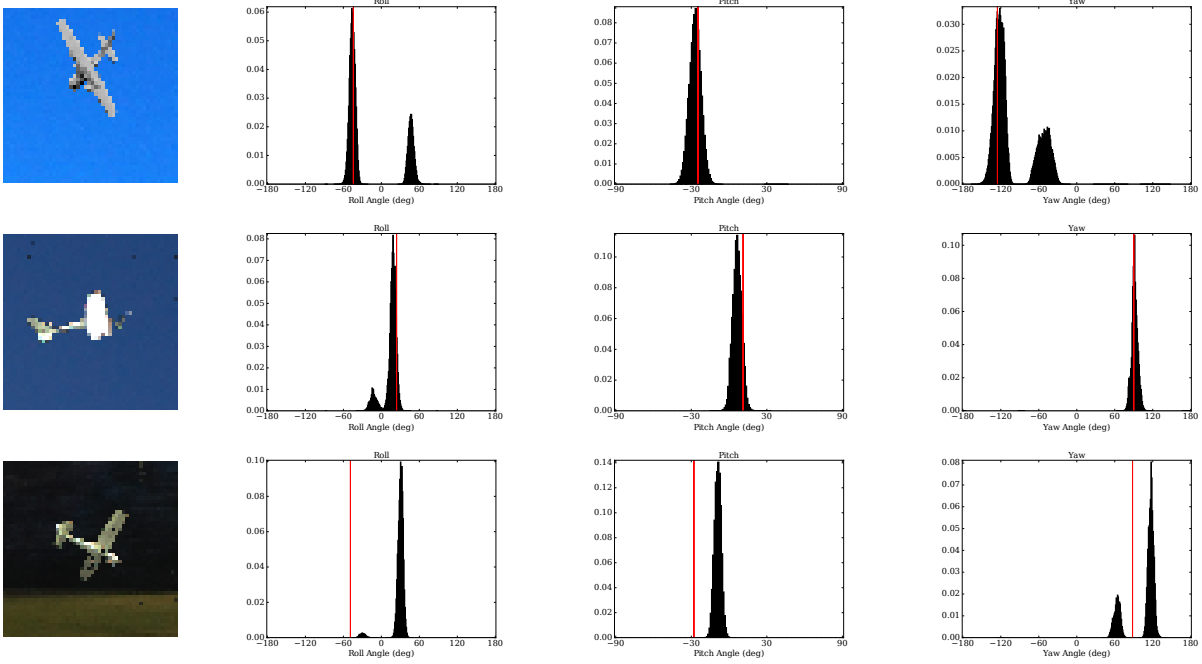


Fig. 3. Top row: Shown is an example of the classification results from the CNN where the red line is the true angle and the black lines are the output of the CNN. Notice the bimodal distributions of the sensed roll, pitch, and yaw angles. This ambiguity is similar to the uncertainty that RC pilots sometimes experience when they can only see the profile of the aircraft. Middle row: Validation image from the motion capture room on the UMX Cessna 182 that produced good attitude measurements. Bottom row: Validation motion capture image that resulted in relatively poor attitude measurements.

angles discretized at one degree increments. The softmax probability $P(\mu_\nu|x)$ is the normalized probability output from the last layer of the network, the output of which is shown in Figure 3. This creates a softmax loss function weighted by the angular distance and preliminary testing showed that it provides a distribution of probability mass around the true value better than using a cross entropy type loss.

The useful output from the network is the normalized probability for each roll, pitch, and yaw (ϕ , θ , and ψ respectively) and is shown in Figure 3. These distributions often are bimodal, which can be expected from a monocular image of an aircraft profile. This bimodal uncertainty is similar to how novice RC pilots often momentarily confuse the orientation of an aircraft. For example, if a pilot can only see the profile of a banking aircraft, it is difficult without prior information to tell if the aircraft is turning toward or away from the pilot. These non-Gaussian uncertainties from the visual sensing were the primary motivation for using a particle filter for state estimation.

C. Motion Capture Validation

The training was validated by creating an entire new validation data-set from a motion-capture room and the realistic looking airplanes shown in Figure 1. In the motion-capture room, the set-up included having motion-capture cameras measure the orientation of the aircraft while a front facing camera captured an image of the aircraft. It was then pivoted to a variety of orientations as the images were captured. Finally, the motion-capture room background was replaced in

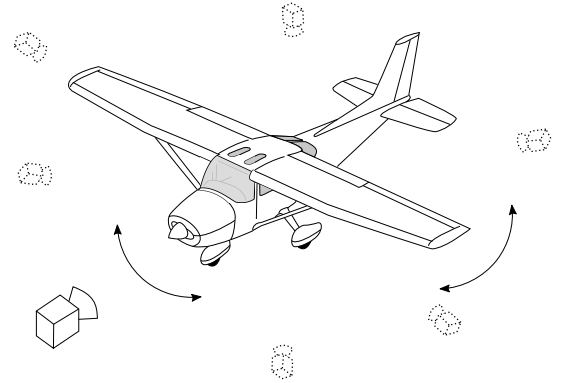


Fig. 4. The motion-capture room set-up allows for rotating the airplane to different orientations. This data provided ground truth of the airplane's orientation for comparison to the CNN's output. Dotted cameras represent motion-capture cameras while the solid-lined camera shows the camera which captures the aircraft image.

the same manner as in the training and testing data. Figure 4 depicts the motion-capture room set-up and Figure 3 includes two examples of the validation results.

These results validate the ability of the CNN to classify the orientation of the aircraft with respect to the camera frame and provide a solid foundation for the complete estimation and control of an aircraft from the third-person perspective. Figure 3 shows two validation images and their respective classifications, one accurate and one inaccurate.

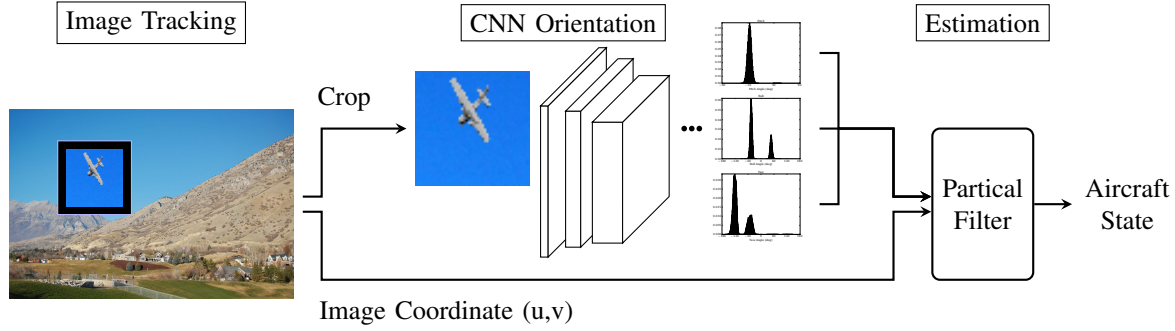


Fig. 5. This final topology of the complete proposed system. Tracking of the aircraft is performed on images of a stationary camera. A 50×50 pixel crop of the aircraft is sent to the CNN for obtaining an attitude measurement. The cropped image is processed through five convolutions and three fully-connected layers. The attitude measurement, in the form of histograms, together with the pixel coordinate of the tracked aircraft are sent to a particle filter. The particle filter uses these measurements to estimate the state of the aircraft.

IV. PARTICLE FILTER ESTIMATION

The filter will provide a complete state estimate of the aircraft's position and orientation and utilize the Markov property to summarize the time history of measurements and states into a current state estimate. We chose to utilize a particle filter because it can accommodate a measurement with an arbitrarily complex uncertainty distribution, where as other estimation techniques (such as the Kalman filter) use an approximately Gaussian distribution [23]. The filter estimates the aircraft's orientation (ϕ , θ , and ψ) and global position in north-east-down coordinates (x , y and z). The full state vector is

$$\mathbf{x} = [x \ y \ z \ \phi \ \theta \ \psi]^T.$$

A. Propagation Model

To utilize a simple propagation model, a number of assumptions were made about how the aircraft was flying. While these assumptions are optimistic, they do capture the general flight characteristics of a non-acrobatic aircraft that is cruising and making banking turns. The first assumption is that the aircraft is flying at a known, constant, forward velocity with small random perturbations. Next, the aircraft was assumed to be flying level, at an altitude with a slow random walk.

If the aircraft is flying level and coordinated, then a simplified coordinated-turn equation, as outlined in [24], can be used to approximate the yaw rate ($\dot{\omega}$) of the aircraft. The equation is

$$\dot{\omega} \approx \frac{g}{v} \tan(\phi) \quad (1)$$

where g is gravitational constant, v is the forward velocity, and ϕ is the roll (or bank) angle.

A propagation model for a planar robot is presented in [23]. Since the aircraft is flying level and the yaw rate can be calculated in (1), we elected to use the model to update the north and east position of the aircraft.

The complete propagation model is

$$\mathbf{x}' = \mathbf{x} + \begin{bmatrix} -\frac{v}{\omega} \sin(\psi) + \frac{v}{\omega} \sin(\psi + \hat{\omega} \Delta t) \\ \frac{v}{\omega} \cos(\psi) - \frac{v}{\omega} \cos(\psi + \hat{\omega} \Delta t) \\ \eta_z \\ \eta_\phi \\ 0 \\ \hat{\omega} \Delta t \end{bmatrix}$$

where η_z and η_ϕ are sampled from a Gaussian distribution. Note that, in this simplified model, the pitch angle does not evolve. The pitch angle will, however, be important for the measurement model perspective transformation described below.

This model, while simple, allows the filter to capture the correlation of the lateral/directional dynamics. Roll angle is correlated with both the north/east position and yaw angle of the aircraft. This coupling allows the measurement of attitude from the CNN to also help observe the position of the aircraft, and conversely, the image coordinate measurements from the tracker to help observe the attitude that was necessary to get the aircraft to its current position.

B. Measurement Models

In this work, the camera is assumed to be at the origin of the global, north-east-down coordinate frame. The camera is also oriented with the optical axis pointing along the global x -axis and the camera x -axis aligned with the global y -axis. The aircraft states are computed with respect to the global reference frame.

Camera images are processed using a mixture-of-Gaussian background subtractor to distinguish the aircraft from the background clutter. Blob detection is then performed on the foreground to inform a simple tracker of the location of the track in the image frame. These image processing algorithms were implemented using the functions provided in the OpenCV vision processing libraries [25]. A image crop of the tracked aircraft is then used as the input to the CNN. Both the CNN output and the image coordinates of the track are used as measurements.

The CNN was trained to perceive the orientation of an aircraft that is close to the optical axis of the camera. The apparent orientation of the aircraft, however, may be very different than the actual global orientation when the aircraft is far from the optical axis.

As an illustrative example, imagine an aircraft moved clockwise in a circle around an observer without changing its global orientation. A camera continuously fixed on the aircraft would observe it yawing to the left, although no yaw motion took place. Similarly a level aircraft pointing away from the observer would appear to pitch down as it moved upward. This apparent orientation of the aircraft must be accounted for when incorporating the orientation measurements from the CNN.

We define an observation axis as a line from the camera center (and global origin) to the position of a given particle. The apparent roll, pitch, and yaw angles can be calculated by applying a transformation consisting of azimuth and elevation rotations that would move the camera optical axis to the observation axis. For each particle in the filter, apparent roll, pitch, and yaw angles are calculated and their weight assigned from the probability density produced by the CNN's softmax output of the orientation angles. This use of the softmax probability makes no assumption about the distribution of the measurement and therefore is able to handle the bimodal distributions, shown in Figure 3, that are produced by the CNN.

Using the camera matrix and distortion parameters the normalized global positions ($\frac{y_e}{x_n}$ and $\frac{z_d}{x_n}$) can be calculated from the pixel coordinates (u, v) provided by the tracker. This can also be used as a measurement to assign a weight to each particle by assuming a Gaussian distribution of the tracker measurement noise.

C. Rao-Blackwellization

After weights are assigned to the particles from both the CNN attitude and image coordinate measurements, the particles are then resampled before they are further propagated. The filter utilizes a low-variance resampler that is described in [23].

We found that the filter works best with between 300-500 particles and that inserting a few particles at random positions and orientations helped prevent particle deprivation problems. Particle deprivation seemed to be especially problematic when the aircraft enters the camera's field of view and the track is first acquired. Particle insertion also improved the robustness of the filter and helped the estimates remain representative of the aircraft true states. To produce the state estimate, the weighted average of the particle states is computed. The true state is then recorded for comparison with true aircraft states.

V. RESULTS

A. Simulation Testing

Initial testing of the complete system, shown in Figure 5, was performed in simulation. The simulation used the ROS/Gazebo simulation tools that were developed as part

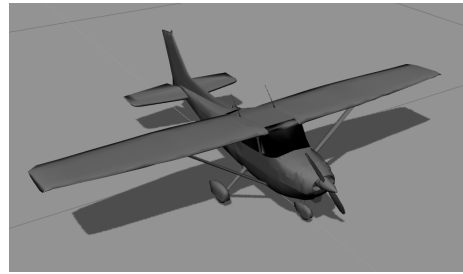


Fig. 6. Simulated model of a Cessna C-172 (scaled to have a .63 m wingspan) used in the Gazebo simulation flight testing.

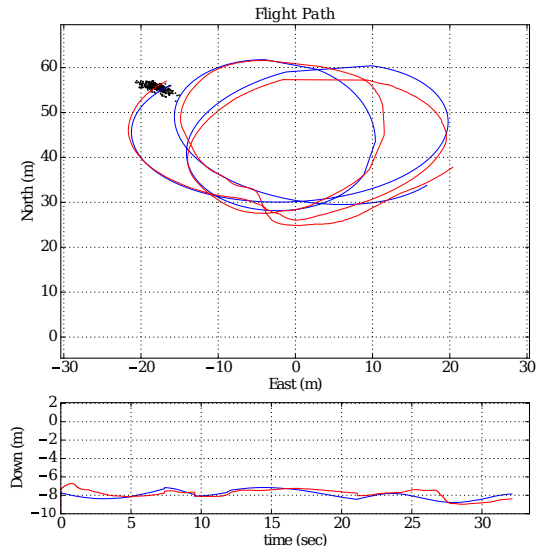


Fig. 7. The trajectory of 32 seconds of simulation flight testing. The true flight path (blue) is plotted against the estimates (red). Particles from the filter are also shown as black dots. The simulated aircraft flew in small continuous/coordinated clockwise turns at a nearly constant altitude.

of ROSplane [26]. The aircraft model was replaced with a Cessna C-172 flying on a blank background. The model can be seen in Figure 6.

To improve the fidelity of the simulation testing, a simulated camera was placed in the simulation environment at the global origin. The camera had a resolution of 1280×960 and had a horizontal field of view of 57 deg. The same image processing was used on the simulated camera image and the imagery from hardware flight testing.

Figure 7 shows the results of testing the proposed CNN and particle filter in the simulation. In the simulation, a controller was used to fly the aircraft in small clockwise circles. The controller was tuned so that the simulation represented the flight ability of a novice pilot. It was also tested to demonstrate that, without the CNN, the propagation model and pixel coordinate measurement would not work. In other words, without the CNN attitude measurements to inform the filter, the estimates would either fail to converge or eventually diverge from the true states.



Fig. 8. Flight testing image (left) and cropped image (right). The crop is sent to the CNN to create a measurement. The original image was 1920×1080 and the crop is 50×50 pixels.

B. Flight Demonstrations

The proposed method was also tested on hardware. The hardware included a stationary camera sitting on a small hill so the aircraft could fly in front (and slightly above or below) of it. The aircraft tested were the aircraft shown in Figure 1. They were also fitted with a small uBlox Global Positioning System (GPS) sensor to record the true position of the aircraft. Images and GPS measurements were recorded and processed with the proposed method in post-process.

Figure 8 shows an example of a recorded image and the aircraft flying in the frame. Also shown is one of the cropped images provided by the tracker and image processing. We note that the aircraft at this distance provides a very poor resolution for the CNN to observe the attitude. This was the primary problem that we encountered in flight testing.

The aircraft were selected because they are light, small, and fly slowly compared to other RC aircraft. This allows the aircraft to remain relatively close to the camera throughout the flight. A Nikon D750 camera with a 18-55mm lens was selected because it had a wide capability and high pixel resolution (full HD). The aircraft was flown by a RC pilot in small tight circles. Even with these efforts, it proved difficult to get enough resolution of the aircraft for the CNN to provide good attitude measurements. In the results shown in Figure 9 the aircraft was flown so close to the camera that it temporarily flew out of the frame, as seen by the dashed lines representing the camera field-of-view. This caused the estimates to temporarily suffer before the track of the aircraft was reacquired and estimates then reconverge.

To produce these results, the filter's airspeed and propagation noise had to be hand tuned to produce the optimal estimation accuracy. Other than this tuning, the same filter produced both the simulation and flight-test results.

VI. CONCLUSION

We draw two conclusions from the above results. First, a CNN can be used to provide attitude measurements (and measurement uncertainties in the form of histograms) of a small fixed-wing aircraft and, second, the measurements can be utilized in a particle filter together with a simple propagation model to estimate the aircraft position.

This work is in contrast to other deep learning methods that focus on end-to-end approaches, where the whole architecture is trained to directly produce a control policy.

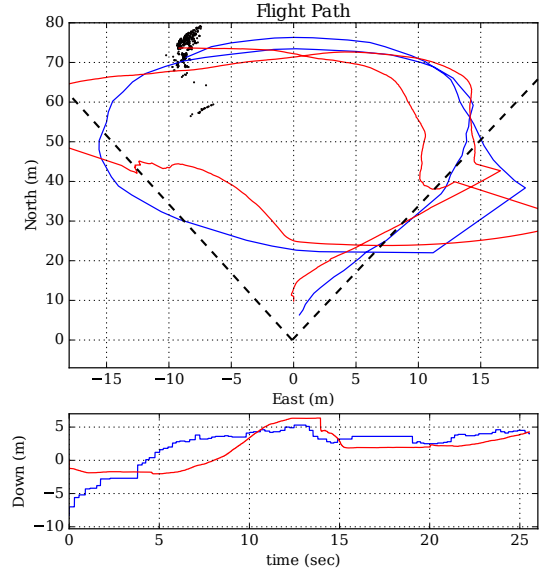


Fig. 9. The trajectory of 26 seconds of flight testing with the aircraft flying counter clockwise circles. The true flight path from recorded GPS (blue) is plotted against the estimate position (red). Particles from the filter are also shown as black dots. Dashed lines show the approximate field-of-view of the camera. The aircraft briefly flew out of frame on both sides which also caused poor position estimates for brief periods before new measurements are allow the estimates to reconverge.

It confines the deep learning to sensing only and allows more classical computer vision and estimation approaches to propagate a aircraft model, compute an aircraft state state, combine measurements, and account for uncertainty.

We consider this work a proof-of-concept that deep learning methods for perception of an aircraft's orientation can be roughly comparable to the ability of a novice RC pilot. The main limitation of this work is demonstrated by the less accurate results from the flight demonstration. We hypothesize that a higher resolution image of the aircraft would lead to a better orientation measurements from the CNN. This could be accomplished with a higher resolution camera, though this would increase the computational burden of image processing for the tracker. Alternatively, a gimbaled camera with a telephoto lens would accomplish the same objective, and would be perhaps more similar to the head and eye movement of an actual RC pilot.

In addition to improving resolution, future work could include a simple controller around the estimator. A less simplistic propagation model would also allow the aircraft to fly dynamically in three dimensions. Furthermore, knowledge of the RC stick inputs or controller commands would inform the propagation model and allow the filter to learn to "trust the sticks," which is similar to a the skills developed by intermediate or advanced RC pilots.

Overall, we believe that using a CNN as a sensor is viable and provides a compelling solution for aiding SUAS estimation. This third-person-perspective approach establishes a way for a CNN to estimate an aircraft's orientation, comparable to a novice RC pilot.

REFERENCES

- [1] D. K. Kim and T. Chen, "Deep neural network for real-time autonomous indoor navigation," *CoRR*, vol. abs/1511.04668, 2015. [Online]. Available: <http://arxiv.org/abs/1511.04668>
- [2] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems (NIPS)*, 2012, pp. 1097–1105.
- [3] J. Schmidhuber, "Multi-column deep neural networks for image classification," *Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3642–3649, 2012.
- [4] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. E. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.
- [5] T. Zeng, R. Li, R. Mukkamala, J. Ye, and S. Ji, "Deep convolutional neural networks for annotating gene expression patterns in the mouse brain," *BMC Bioinformatics*, vol. 16, p. 147, 2015.
- [6] D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. Van Den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot *et al.*, "Mastering the game of Go with deep neural networks and tree search," *Nature*, vol. 529, no. 7587, pp. 484–489, 2016.
- [7] J. Wang and C. Yuan, "Facial expression recognition with multi-scale convolution neural network," in *Pacific Rim Conference on Multimedia*. Springer, 2016, pp. 376–385.
- [8] G. Hinton, L. Deng, D. Yu, G. Dahl, A.-R. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. Sainath, and B. Kingsbury, "Deep neural networks for acoustic modeling in speech recognition," *IEEE Signal Processing Magazine*, vol. 29, no. 6, 2012.
- [9] L. Deng, G. E. Hinton, and B. Kingsbury, "New types of deep neural network learning for speech recognition and related applications: an overview," in *Proceedings of International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2013, inproceedings, pp. 8599–8603.
- [10] O. Abdel-Hamid, A.-R. Mohamed, H. Jiang, L. Deng, G. Penn, and D. Yu, "Convolutional neural networks for speech recognition," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 22, no. 10, pp. 1533–1545, Oct. 2014.
- [11] G. E. Dahl, D. Yu, L. Deng, and A. Acero, "Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 20, no. 1, pp. 30–42, 2012.
- [12] Z. Xu and J. Huang, "Efficient lung cancer cell detection with deep convolution neural network," in *International Workshop on Patch-based Techniques in Medical Imaging*. Springer, 2015, pp. 79–86.
- [13] L. Zhuo, L. Jiang, Z. Zhu, J. Li, J. Zhang, and H. Long, "Vehicle classification for large-scale traffic surveillance videos using convolutional neural networks," *Machine Vision and Applications*, pp. 1–10, 2017.
- [14] F. Zhang, J. Leitner, M. Milford, B. Upcroft, and P. I. Corke, "Towards vision-based deep reinforcement learning for robotic motion control," *arXiv*, vol. 1511.03791, 2015.
- [15] A. X. Chang, T. Funkhouser, L. Guibas, P. Hanrahan, Q. Huang, Z. Li, S. Savarese, M. Savva, S. Song, H. Su, J. Xiao, L. Yi, and F. Yu, "ShapeNet: An Information-Rich 3D Model Repository," Stanford University — Princeton University — Toyota Technological Institute at Chicago, Tech. Rep. arXiv:1512.03012 [cs.GR], 2015.
- [16] B. S. Chiel and J. Baillieul, "Visual GPS-denied multi-agent localization & terrain classification," in *2018 IEEE Aerospace Conference*. IEEE, 2018.
- [17] G. J. Ellingson, D. Wingate, and T. McLain, "Deep visual gravity vector detection for unmanned aircraft attitude estimation," 2017.
- [18] C. M. Bishop, "Pattern recognition and machine learning," *Springer*, 2006.
- [19] D. Schulz, W. Burgard, D. Fox, and A. B. Cremers, "Tracking multiple moving targets with a mobile robot using particle filters and statistical data association," in *Robotics and Automation, 2001. Proceedings 2001 ICRA. IEEE International Conference on*, vol. 2. IEEE, 2001, pp. 1665–1670.
- [20] G. Tong, Z. Fang, and X. Xu, "A particle swarm optimized particle filter for nonlinear system state estimation," in *Evolutionary Computation, 2006. CEC 2006. IEEE Congress on*. IEEE, 2006, pp. 438–442.
- [21] D. E. Clark and J. Bell, "Multi-target state estimation and track continuity for the particle phd filter," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 43, no. 4, 2007.
- [22] H. Su, C. R. Qi, Y. Li, and L. J. Guibas, "Render for CNN: Viewpoint estimation in images using CNNs trained with rendered 3D model views," in *The IEEE International Conference on Computer Vision (ICCV)*, December 2015.
- [23] S. Thrun, W. Burgard, and D. Fox, *Probabilistic robotics*. MIT press, 2005.
- [24] R. W. Beard and T. W. McLain, *Small Unmanned Aircraft: Theory and Practice*. Princeton University Press, 2012.
- [25] G. Bradski, *Dr. Dobb's Journal of Software Tools*.
- [26] G. Ellingson and T. McLain, "ROSplane: Fixed-wing autopilot for education and research," in *Unmanned Aircraft Systems (ICUAS), 2017 International Conference on*, 2017.