This article has been accepted for publication in IEEE Intelligent Systems. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/MIS.2025.3607236

# Aura: An Automated System for the Real-Time Evaluation of Flight Maneuver Performance

Mahdi Al-Husseini, Joshua Barnett, Joseph Divyan Thomas, Tony G. Chen

***Abstract*—The automated evaluation of flight maneuver performance in civilian and military aircraft enhances pilot proficiency and aircraft safety. In this article, we present Aura, an in-aircraft flight maneuver training system that provides real-time performance feedback to pilots. Aura consists of sequential flight data capture, flight data analysis, and flight data visualization modules. The flight data capture module uses a pipeline of computer vision techniques and object detection algorithms to collect flight data optically. The flight data analysis module applies a transformer-based classifier network, trained on a custom dataset, to identify flight maneuvers in real-time. The flight data visualization module displays processed data to pilots on a hardware display in a task-specific layout. Aura does not need a physical interface with aircraft avionics, thereby circumventing data access issues in military aircraft. Aura is ground tested using UH-60M Black Hawk helicopter flight simulator recordings and is flight tested for several hours in a Cessna 172S G1000 to validate effectiveness.**

## I. Introduction

Rapid and reliable in-flight data collection and analysis tools can significantly enhance flight training [1]–[3], flight test and evaluation [4]–[6], flight safety [7]–[9], and flight maintenance [10]–[12]. Evaluating pilot operational performance, often through flight maneuvers and their objective standards, is critical to determine pilot proficiency in a given aircraft or for a particular mission [13]. Flight envelope expansion procedures may be expedited by evaluating collected data in-flight, discarding data-points that do not conform to requirements, and repeating trials where necessary [14]. Loss of control in-flight (LOC-I) may be prevented by analyzing flight data and helping detect stalls and spins before they occur, or by alerting pilots of their attitude, torque/power, and altitude during inadvertent entry into instrument meteorological conditions (IIMC) [15]. Certain maintenance tasks may also benefit from in-flight assessment against maintenance standards [11].

Many modern aircraft avionics systems collect, store, and transmit flight data via a quick access recorder (QAR). QARs provide easy access to raw flight data and enable real-time performance, safety, and maintenance monitoring [16], but are not legally mandated for use. Further, aircraft with older avionics and the majority of military aircraft lack QARs or similarly accessible systems [17]. This limits many pilots' ability to access raw data in flight as well as the safety and performance monitoring tools that require it. Additionally, few flight training vehicles exist to enable rigorous self-evaluation in-flight. A real-time evaluation tool, as opposed to post-flight data analysis, enables pilots to quickly adjust between subsequent maneuvers rather than between flights – resulting in faster learning and enhanced outcomes.

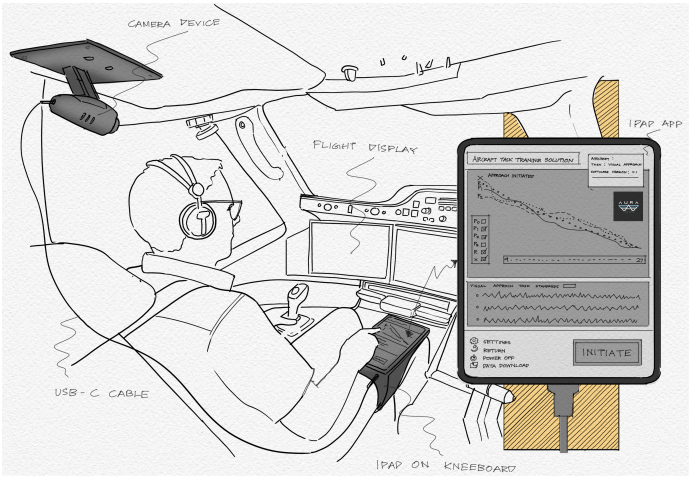



Fig. 1: Representation of Aura in the cockpit.

We develop, deploy, and evaluate Aura (Fig. 1), an in-aircraft flight maneuver training system applying machine learning and computer vision techniques (Fig. 2). Aura is designed for both military and civilian aircraft and facilitates non-obtrusive, non-intrusive, and user-friendly data collection from analog or digital flight instrumentation. Aura further provides pilots real-time flight maneuver performance feedback to enhance in-aircraft training and can reasonably support a range of in-flight maintenance, flight test and evaluation, and safety of flight tasks.

Aura has sequential flight data capture, flight data analysis, and flight data visualization modules [18]. To demonstrate applicability to military aircraft, Aura is ground tested using video from UH-60M Black Hawk helicopter flight simulators. To validate effectiveness, Aura is flight tested for several hours in a Cessna 172S conducting commercial flight maneuvers. To our best knowledge, Aura is the first all-in-one example of an in-aircraft, real-time, flight maneuver performance training system that can obtain a wide range of flight instrumentation and engine performance data without directly interfacing with aircraft avionics. Our contributions follow:

1) A flight data capture module applies machine learning and computer vision techniques to identify and select relevant flight instruments from an instrument panel, then obtains and aggregates flight data from those instruments. This enables a single, well-positioned camera to serve as an aircraft flight data recorder without needing to interface with aircraft avionics.

This article has been accepted for publication in IEEE Intelligent Systems. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/MIS.2025.3607236

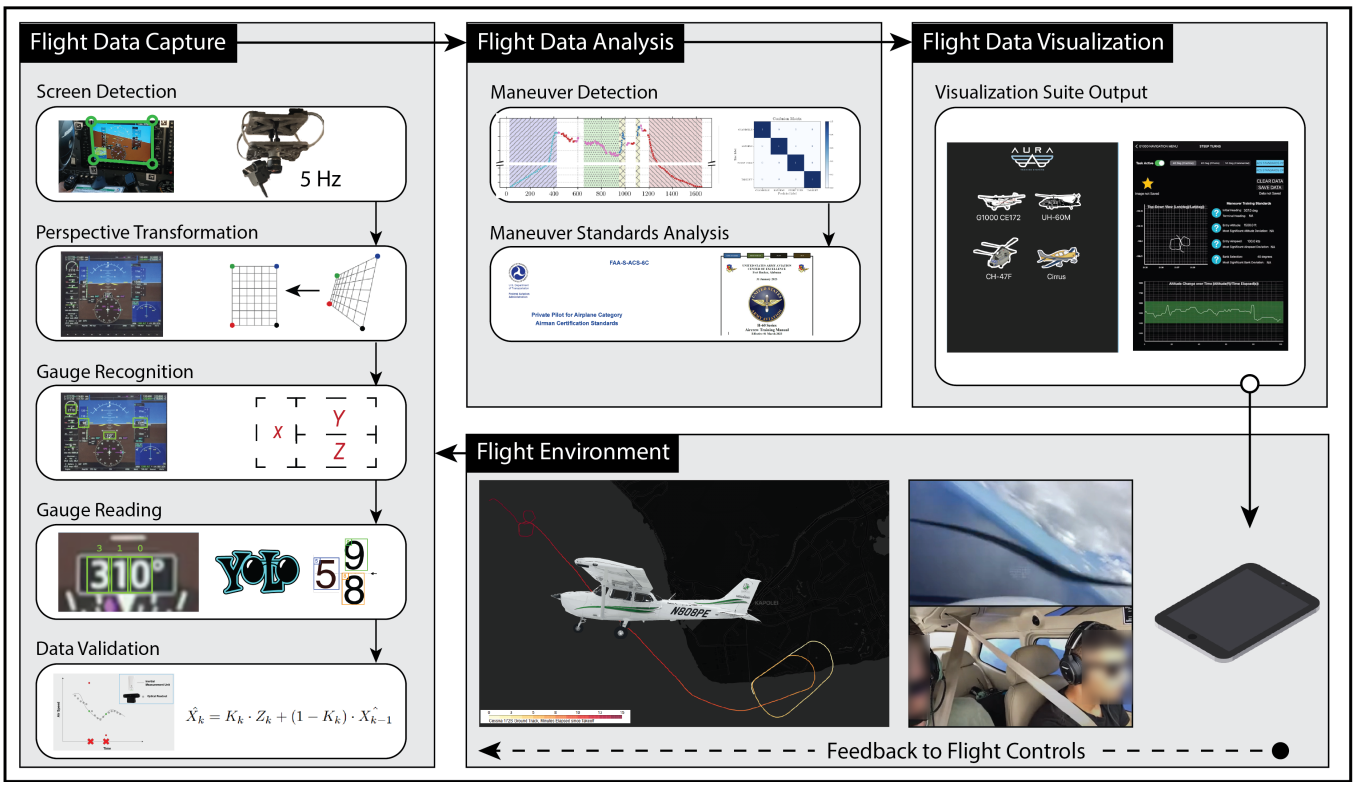



Fig. 2: Aura system architecture includes flight data capture, flight data analysis, and flight data visualization modules. The flight data capture module applies computer vision techniques to transcribe flight instrumentation data. The flight data analysis module applies transformers to classify flight maneuvers. The flight data visualization module displays performance data in a pilot-friendly and maneuver-specific format for efficient knowledge transfer in flight. © Mahdi Al-Husseini 2023

2) A flight data analysis module uses transformers to identify the flight maneuver being conducted. Flight maneuver performance is then evaluated by comparing the processed data against civilian or military maneuver performance standards. We use a digital flight simulator to generate a training dataset of 101 labeled flight maneuvers, which we provide alongside in-flight test data.[1]
3) A flight data visualization module presents maneuver performance data to the pilot in real-time on an in-aircraft hardware device with a custom graphical user interface.

## II. Background

### A. Computer Vision

Computer vision techniques have recently been used for in-flight applications to include assessing abnormal pilot behavior, performing pilot hand detection in complex scenes, transcribing analog and digital aircraft instrumentation, and detecting external airspace traffic. In [19], a custom attention mechanism is used with an established deep neural network-based object detection and localization algorithm (YOLOv4) to identify abnormal pilot behaviors and minimize associated safety risks. Another approach to pilot monitoring focuses on hand movements and applies image segmentation followed by key point and contour extraction to identify the locations of pilots' hands during flight tasks [20]. Object detection networks have been used to read radial dials, with an image segmentation step further applied to isolate the dial needle and attain an approximate instrument value [21]. Recent work in [3] relies on deep artificial neural-network (ANN)-based optical character recognition (OCR) software to read digital flight displays, which would be challenging to use for vertical tape-type displays encountered in the present work. Finally, numerous patents and patent applications employ cameras in and around the cockpit to identify air traffic and obstacles and issue alerts to prevent collision avoidance and controlled flight into terrain [22], [23]. Object detection, image segmentation, ANNs, and alerting infrastructure all feature heavily in our flight data capture, analysis, and visualization modules.

### B. Maneuver Classification

A large body of maneuver classification literature has emerged in the last two decades, initially dominated by simple, rules-based classification methods and trending towards machine-learning and deep ANN-based approaches. An early approach considered a series of features of a reference maneuver and compared the distance against that of features

[1]Source code and data available at https://github.com/jbarnett8/AuraFlight

from a queried maneuver [24]. Unsurprisingly, this process involves significant calibration and fails to generalize well to other maneuvers or aircraft. A more sophisticated approach in [25] develops a library of maneuver features used as a series of rule-based classifications. Similar research in [26] considers specific maneuvers for H-60 series Navy helicopters. These rules-based approaches suffer from fragility introduced by a simplistic scheme unable to address edge cases. More recent work applies conventional machine learning classification algorithms such as logistic regression classifiers and support vector machines (SVM) to label segments of time series data obtained from an aircraft [27]. Results are then compared to an ANN-based approach. Unfortunately, the conventional machine learning methods largely out-performed the ANN-based methods, meaning that a suitable dataset where conventional methods failed was not explored, if such a case exists.

To address the limitations in the existing literature, we use recurrent neural network (RNNs) models to classify time series data. Although this method is more data-intensive than traditional machine learning classifiers, it is likely to generalize better to other maneuvers and airframes with smaller training sets than were required to generate the initial model. Pre-training is an effective accuracy enhancing and cost-saving method widely used in deep ANN-based applications; it is likely exploitable here as well. Coupling this advantage with their ability to classify sequences with arbitrary length makes RNNs a promising approach for maneuver classification.

### C. Automated Flight Maneuver Performance

The automated evaluation of flight maneuver performance enhances flight simulator training and post-flight debriefings. Both applications support self-learning by providing pilots with comparative flight maneuver standard data and tailored feedback. Yang et. al. introduce a machine learning supported framework for training pilots in flight simulators [3]. An offline mode uses datasets from expert pilots to train a machine learning algorithm to predict control signals that are then compared with trainee pilot actions. Pilot feedback is generated based on the difference. CloudAhoy is a post-flight debriefing tool that collects and analyzes data from the Garmin G1000 avionics suite and other data input sources to, in part, assign a pilot flight maneuver performance scores [28]. CloudAhoy provides pilots with an easily accessible post-flight visualization suite, and can identify student "problem areas". Zhang et. al. examine the correlation between aircraft operational status indicators and pilot performance, and are able to adequately evaluate overall pilot skill level using a one-dimensional convolutional neural network that considers QAR parameters [29].

## III. Flight Data Capture, Analysis and Visualization

### A. Optical Flight Data Capture

The flight data capture hardware assembly consists of (1) a vibration-dampening mounting structure with battery bank carriages to attach the assembly stably to the inside of the cockpit; (2) a Raspberry Pi 4 to act as the host device; and (3) an OAK-D from Luxonis as both a high-resolution image capture and hardware accelerator device for ANN-based computer vision algorithms. Restricting most image-based computation to the hardware accelerator aboard the OAK-D allows for real-time data capture, achieving roughly 5 frames-per-second. Performing all components of the computer vision pipeline on the edge is important in this case because it does not require a reliable, high-bandwidth connection to the internet where more powerful models may be able to deliver more accurate results at the price of speed, privacy, and security.

The data capture process is non-obtrusive, which requires that it be robust to variations in mounting and calibrating the data capture assembly, which leads to images captured from differing perspectives. Performing inference on images with differing perspectives from the training set can be challenging for ANN-based computer vision techniques. As opposed to conditioning these components of the data capture system directly with images from different perspectives, we instead apply a four-point perspective transform so that all inputs have roughly the same perspective. We use functions provided in the `OpenCV` package [30] to pre-processing images prior to applying the ANN-based computer vision models. This has at least four major benefits: (1) it increases the explainability of the model, where fewer of the operations are abstracted to a neural-network that could be handled by analytical approaches; (2) it requires less training data, where only head-on perspective data capture is required to obtain a suitable training set; (3) it produces a more specific model that requires a smaller number of trainable weights, which decreases training time, likelihood of overfitting, and inference time; and (4) it compliments algorithms that can only provide bounding boxes in the form of rectangles, which have little to no information about the orientation of the detected object. For more reliable screen detection, the corners of the flight display are marked using AprilTags [31], which are visual fiducial markers used to identify position, orientation, and identity with respect to the camera position.

One ANN-based computer vision algorithm incorporated into the data capture system is a YOLOv4 network [32], which is normally used for object detection. This makes it an obvious choice for isolating regions of interest in an image to instruments with relevant flight data [33]. From a given image, all of the instruments can be identified and interpreted independently in such a way that incorporates their unique designs.

One differentiating feature of our approach is the use of this same type of network specifically for character recognition for analog displays. Certain instruments require knowledge of the relative vertical location of numbers indicating the reading. Take for example a vertical tape-type display, which is often used to report airspeed and/or altitude in the instrument panel cluster. To correctly interpret the reading on the dial, an algorithm must know where the numbers are spatially located, where the index line is (where to read the measurement), and how to handle non-centered readings (where, for example, a

This article has been accepted for publication in IEEE Intelligent Systems. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/MIS.2025.3607236

digit may not rest directly on the index line). Importantly, the YOLOv4 can provide the complete location of all the numbers within a region of interest assuming it has been properly fine-tuned, which can enable a relatively simple algorithm to then produce an accurate value of the gauge's reading.

This network used in the data capture assembly software pipeline is trained starting from an existing checkpoint of the original YOLOv4 network, which is publicly available. This checkpoint is fine-tuned using another publicly accessible dataset consisting of over 600,000 labeled digits [34]. Finally, the model undergoes another level of refinement where images of a Garmin G1000 display while in-flight are recorded, resulting in around 2000 additional labeled digits. Reserving 10% of these in-flight digits for testing shows that the final model is capable of around 95% classification accuracy.

For use in-flight, the YOLOv4 model is serialized for inference using the built-in hardware accelerator aboard the OAK-D camera. Unifying the pipeline—including screen detection, the perspective transform, and gauge reading—all using the visual processor unit allows the pipeline to achieve roughly 5 Hz update frequency. This is useful even in the limited demonstration in the present work because the Garmin G1000's built in logging capability only captures flight data at a rate of 1 Hz. This rate may be insufficient for capturing maneuvers with features spanning a smaller time scale.

### B. Data Filtering

The data capture assembly will provide a group of labeled number boxes for a given gauge. Using the pre-defined index line shown with the arrow in Fig. 3, an estimation of the reading can be calculated based on the inferred place value as well as the relative position of the boxes above or below the index line. For example, the gauge in Fig. 3 should read somewhere between 58 and 59, but would only be 58 if the 8 box is sliced in half by the index line. For practical purposes, however, the gauge reading is taken to be a whole number. The reason for this is the variability in the detection box size and centering, which can often be inconsistent enough to introduce errors comparable to the additional precision we would otherwise try to infer.

Additionally, sometimes even when the bounding boxes are correct, the labels can be incorrect. This is especially difficult to handle in the case where the most significant digit changes. A simple filtering algorithm is implemented to ensure that outlier measurements are ignored. This is accomplished by: (1) rejecting obviously nonphysical measurements such as headings outside the range 1° to 360° and airspeeds much greater than the maximum specifications of the Cessna 172S; and (2) defining a maximum rate of change for each flight indicator and ignoring values that violate this maximum rate of change compared to the most recent valid measurement. These maximum rates are hyper-parameters we set to 25 °/s, 5 kn/s, and 20 ft/s for heading, airspeed, and altitude, respectively. In either case for rejection, missing values are inferred retroactively using linear interpolation after a valid measurement occurs.

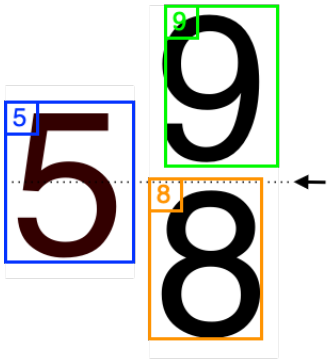
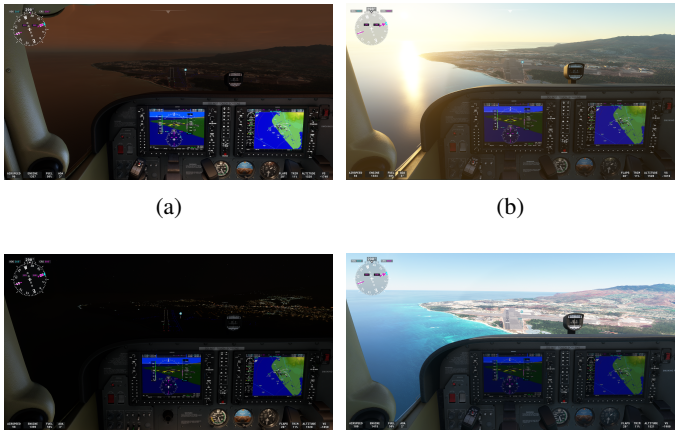



Fig. 3: Illustration of labeled boxes from a vertical tape display obtained from the data capture pipeline. The dashed line indicates the index line, where a measurement is supposed to be taken visually. Using the location and label of bounding boxes, the reading of the gauge can be inferred while taking into account potential errors in bounding box location and label.



(a) (b)

(c) (d)

Fig. 4: Digital flight simulation data recordings and video capture for a Cessna 172 G1000 executing an approach into Kalaeloa airport (PHJR) in varying environmental conditions: (a) sunset with an overcast cloud layer, (b) sunrise with clear skies, glare present, (c) nighttime with scattered clouds, and (d) high noon with clear skies.

### C. Flight Maneuver Recognition

Our classifier is a Transformer-based network leveraging both self- and cross-attention [35] as this architecture is suitable for classification of time series data. To train such a network to recognize specific maneuvers, a labeled dataset of flight time series data is required. Unfortunately, to our knowledge no such publicly available dataset exists. To address this, we generate a custom dataset comprised of several hours of Microsoft Flight Simulator (MFS) digital flight maneuver data runs, resulting in 101 manually labeled maneuvers spanning

This article has been accepted for publication in IEEE Intelligent Systems. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/MIS.2025.3607236

20 different flights. MFS flight run video was recorded to help evaluate the flight data capture process. As shown in Fig. 4, a Cessna 172 G1000 matching that flown during actual flight trials is piloted in various environmental conditions. Simple flight maneuvers preformed include takeoffs and landings. Complex flight maneuvers performed include steep turns and chandelles. Maneuvers were initiated at various altitudes and aircraft headings. MFS flight trials began at airports to include: PHNL (field elevation: 13 ft), PHNG (24 ft), PHJR (30 ft), PHHI (837 ft), PHNY (1308 ft), and PHSF (6190 ft). The collected maneuver data was substantially affected by aircraft performance limitations caused by higher altitudes at PHNY and PHSF.

The dataset is constructed using the comma separated value file output of MFS from each individual flight trial, 20 flights in total. Because one of the goals is to compare with real flight data, the synthetic data from MFS is interpolated to the same 1 Hz logging rate of the Garmin G1000 used to collect our real-world ground-truth data. Out of these 20 flights, 17 are reserved for training, 3 for validation, and 3 for testing. Each of the flights are then further subdivided into their constitutive maneuvers based on the manually applied labels, discarding unlabeled portions of each flight, such that a single sample from a given dataset will be a tuple consisting of: (1) a contiguous segment of the time series flight data and (2) a single maneuver label. The data is further augmented by repeatedly sub-sampling the complete maneuvers from the start of the maneuver to a random time at least 10 seconds from the beginning of that same maneuver but no later than the end of the same maneuver. Repeated sub-samples, i.e. sub-samples whose ending index matches a previous sub-sample, are discarded. This sub-sampling explains why, for example, that the number instances of flight maneuvers in Fig. 6 are large compared to reasonable expectation for 3 full flights. Finally, although the synthetic dataset has a large number of useful features, only altitude, heading, and airspeed are used as these are the indicators captured by the camera system during the real-life test.

The transformer-based classifier network schematic is shown in Fig. 5 which shows the flow of the flight data segments towards the flight segment label. Following to the right side of the diagram, the sequence is biased to begin at zero by subtracting the first time point from the entire sequence. A lifting layer (i.e. multiplying the input by a tall matrix) increases the dimension of the feature and then passes through a multi-head self-attention layer. Following from the left, the time derivative of the flight data segment is approximated through the first-order forward difference scheme and similarly passes through a lifting layer to increase the feature dimension. The cross-attention output is computed using the lifted time derivative values and the output of the self-attention layer. Finally, the final time step in the sequence output by the cross-attention layer is extracted, passed through a multi-layer perceptron (MLP), and a softmax function is applied to obtain a probability distribution.

The model is trained using an Adam optimizer with a class-weighted cross-entropy loss function, which accounts for class imbalances in the dataset. The model trained for 41 epochs and we use the checkpoint with the best validation loss. The results of the test dataset are shown in Fig. 6, indicating that the model is able to perfectly classify all the maneuvers performed in each of the three flights.

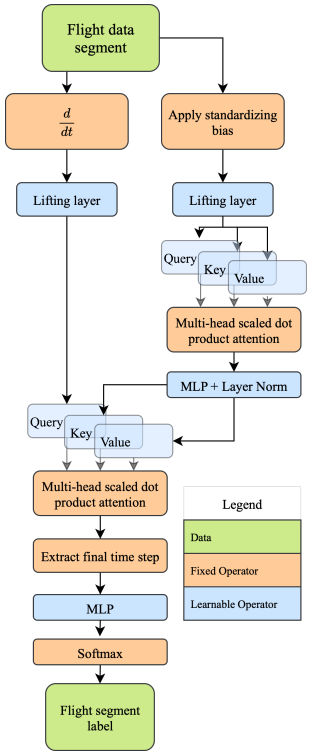



Fig. 5: Schematic of transformer-based classifier network architecture. Layers are colored according to their function.

The model is further evaluated for its ability to classify on-the-fly by feeding it the full flights–not just the maneuver portions–of the test dataset. A sliding window of the full flight with a maximum sequence length of 30 seconds in 10 second intervals is passed to the classifier to label 10 seconds of data at a time. Only labels with at least 80% confidence are accepted. The results of this test are shown in Fig. 7, which demonstrates the ability of the model to correctly classify the labeled portion of the test dataset in all cases.

### *D. User Interface Design*

Thirteen certified general aviation pilots and sixteen rated military aviators provided constructive feedback on six flight maneuver pages associated with two types of aircraft. An iOS application was written in Swift and developed in XCode for use on an iPad affixed to an aviation kneeboard. The application includes an aircraft selection menu, maneuver selection sub-menus, and maneuver analysis pages, as shown in Fig. 8. Four maneuver analysis pages were developed for the UH60-M Black Hawk helicopter: visual meteorological takeoff (VMC) takeoff, VMC approach, roll-on landing, and autorotation. Each UH60-M maneuver analysis page visualizes

This article has been accepted for publication in IEEE Intelligent Systems. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/MIS.2025.3607236

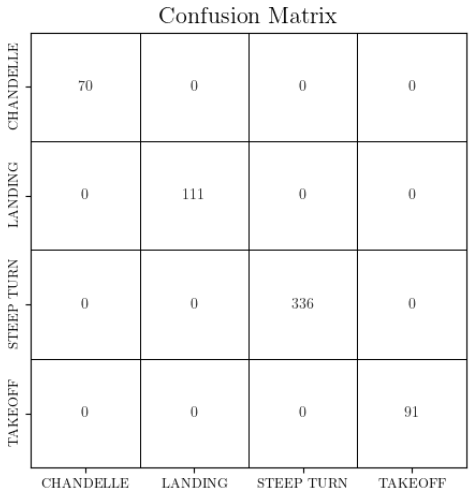



Fig. 6: The best-validation-loss checkpoint transformer-based classifier shows perfect performance labeling isolated full and sub-sampled maneuvers for the test dataset, consisting of 3 flights.

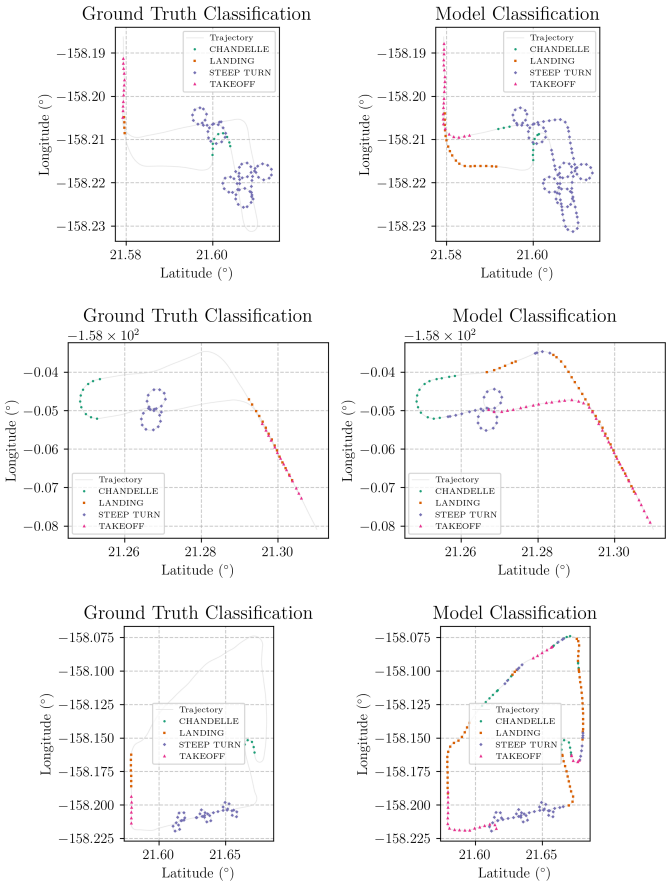



Fig. 7: On-the-fly labeling of the full test dataset using a sliding window of 30 seconds labeling 10 seconds at a time for plots shown on the right. Plots on the left show the ground truth maneuver labels.

flight maneuver standards from the Army Aircrew Training Manual (ATM). Two maneuver analysis pages were developed for the Cessna 172S: steep turns and chandelles. The 172S maneuver analysis pages are designed to illustrate flight maneuver standards in the Airman Certification Standards (ACS). We highlight the Cessna 172S steep turn maneuver analysis page for discussion due to its multiple quantitative ACS standards involving airspeed, heading, altitude, and bank angle. Each maneuver analysis page is designed to be reviewed within twenty seconds. Twenty seconds was determined to be enough time to gain insights from a given maneuver analysis page and yet not so much as to cause substantial flight training delays. A second pilot should maintain aircraft control while the pilot who conducted the evaluated maneuver reviews his performance. Alternatively, should the maneuver be conducted in a traffic pattern near a runway or helipad, a single pilot aircraft should be safely on the ground prior to reviewing performance.

We balance the need to provide concise critical performance highlights quickly in-flight with in-depth contextualizing information to be reviewed post-flight. A task active button (1) supports manual activation of flight data recording, processing, and visualization for a given task. The analysis section is composed of maneuver training standards with deviation data and go/no go evaluations (2) and two-dimension position and altitude graphs with color windows specific to the evaluated maneuver (3). If the pilot is unable to review the maneuver analysis page in-flight, they may save a still image of the maneuver page to the device camera roll (8) and/or save a corresponding raw data file (4) for review at a later time. Visualized data may be cleared (5) from the maneuver analysis page at any time, effectively serving as a page reset. ATM or ACS documentation may be retrieved directly (6) from the maneuver analysis page. Many maneuvers, including steep turns, have different performance standards for certain segments of the aviation population. The pilot may select the appropriate standards for a given maneuver (7) accordingly.

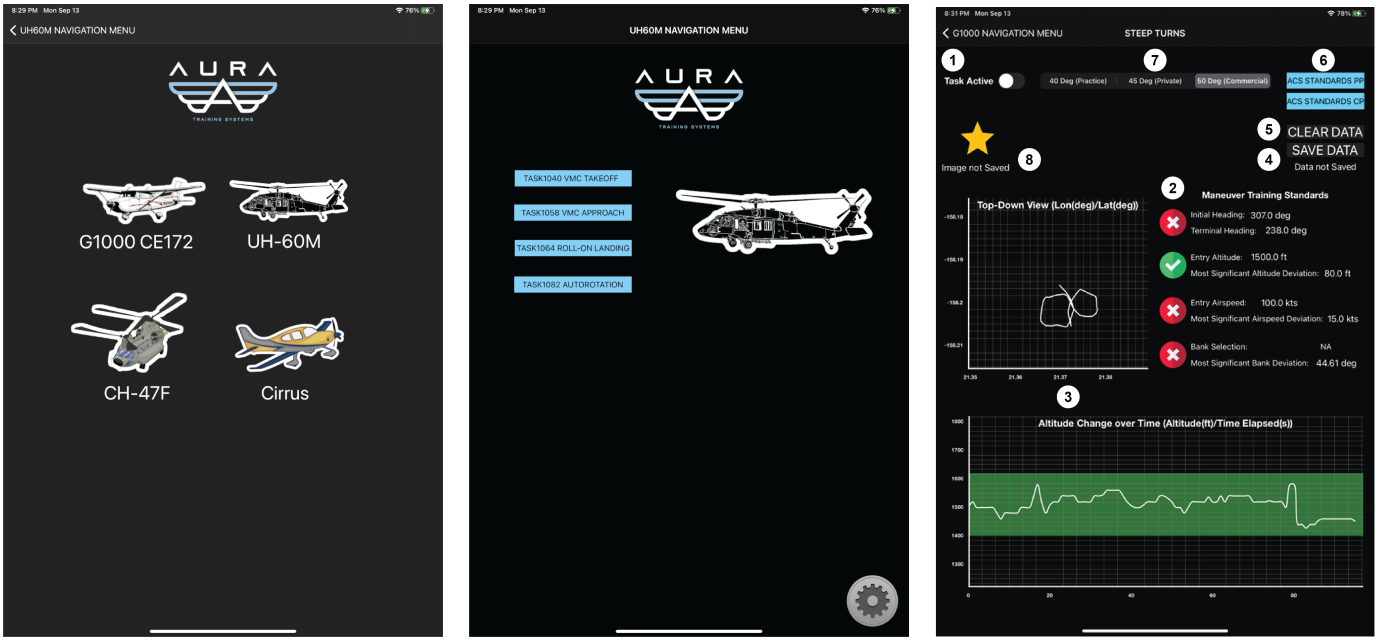
## IV. Experimental Design and Testing Methods

Aura system is first validated and tested in simulation with prerecorded video data from different flight simulators. Then, the system is tested in a real-world scenario using a Cessna 172S aircraft. In the following subsections, we detail the experimental setup along with the results and lessons learned from these simulation and actual flight testing.

### *A. Video Replay and Simulation Testing*

As shown in Fig. 9, Aura was tested using prerecorded UH-60M Black Hawk helicopter footage taken from digital and physical flight simulators. The UH-60M Black Hawk avionics suite consists of four multi-functional displays, each of which may be selected from a set of pages to include a primary flight display, an engine indicating and crew alerting system, a tactical map, and a maintenance menu. Although capturing data from the engine indicating and crew alerting system screen would be valuable for certain maintenance tasks

This article has been accepted for publication in IEEE Intelligent Systems. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/MIS.2025.3607236




(a) (b) (c)

Fig. 8: Screenshots of (a) the aircraft selection menu, (b) a maneuver selection sub-menu, and (c) a maneuver analysis page, from the Aura iOS application designed for in-flight use. (c) analyzes a steep turn maneuver to commercial standards executed during actual flight trials.

and emergency procedure training, we focus on the primary flight display. The primary flight display includes a barometric altimeter, radar altimeter, attitude indicator, compass, airspeed indicator, vertical speed indicator, power pod, fuel indicator, and other miscellaneous secondary instrumentation. We program Aura to analyze the pre-recorded video inputs in place of the live camera stream. The screen detection, perspective transformation, gauge recognition through image segmentation, and gauge reading through object detection steps can be seen in the bounding box and data reading overlays on the right side of the top image in Fig. 9. Aura seamlessly captured the UH-60M engine torque, airspeed, altitude, radar altitude, and heading data across all prerecorded videos.

### *B. Flight Testing*

The Aura end-to-end pipeline was evaluated across several one-hour flights in Hawaii, Alabama, and California during which commercial flight maneuvers were repeatedly performed by a certified flight instructor in a Cessna 172S with a Garmin G1000 electronic flight instrumentation system. Fig. 10 depicts the ground track of a Cessna 172S conducting steep turns after departing from PHJR. Colored fiducial markers/AprilTags were placed in the corners of both primary flight displays to assist with screen detection. The stabilization rig with a camera was fixed to the aircraft cabin ceiling midway between the two front seats. Secondary cameras within the cockpit were attached to the glare shield and windshield to collect additional footage from outside and inside the aircraft respectively. Aura successfully and reliably collected, processed, and visualized chandelle and steep turn maneuver data on the operatively coupled tablet (iPad Air 4th Generation) secured to the pilot's kneeboard. Several lessons were learned during deployment.

**Glare and Vibration**

Environmental factors during flight tests occasionally disrupted the flight data capture process. Intermittent glare on the primary flight displays during certain times of day resulted in partial or full omission of flight parameters considered during flight data analysis. In both the partial and full case, data during the glare interval was automatically filtered out. As previously discussed in the data validation step, fusing and filtering data from supported sources to include the tablet may minimize glare disruptions. Aircraft vibration is a natural consequence of powered flight, and has been demonstrated to exceed $0.25\,\mathrm{m/s^2}$ in a climbing Cessna 172R [36]. Natural aircraft vibration was demonstrated to substantially affect the internally mounted cameras without shock-absorbing fixtures, to an extent preventing data capture. To address this issue, a camera vibration isolation mount was designed and built, shown in Fig. 11. The mount consists of two plates, connected to each other through four stainless steel cables, preloaded in the bending configuration, and four elastic cords placed in tension. This design and configuration are typical in camera gimbal vibration isolators and is proven to be effective in testing. The mounting hardware constructed was added to later flights and proved sufficient for data capture purposes.

**Minimizing Pilot Distraction with Automation**

Although the user interface was designed to minimize

This article has been accepted for publication in IEEE Intelligent Systems. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/MIS.2025.3607236


(a)

(b)

Fig. 9: UH-60M Black Hawk helicopter avionics as seen in (a) a physical flight simulator and (b) a digital flight simulator.

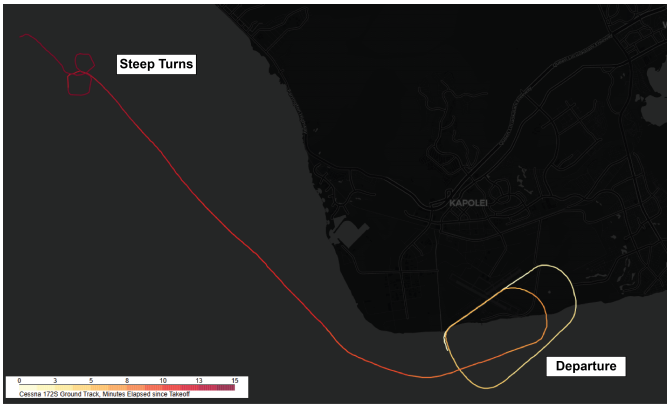



Fig. 10: Ground track showing the first leg of one of several flight tests during which commercial flight maneuvers were performed in a Cessna 172S.

interaction during flight, there remain several opportunities to further reduce pilot distraction. Data collection may begin automatically at the start of a maneuver, or once clearing turns are recognized, as determined by the underlying maneuver classifier. Similarly, data collection may be automatically concluded following the detected completion of a maneuver. Mechanisms for auto-saving the raw data and screenshot, with maneuver designation and timestamp labeling, may then

Fig. 11: Setting up the shock-absorbing fixture for the cockpit camera in a Cessna 172S. © Mahdi Al-Husseini 2023

be incorporated into the iOS application. The pilot currently selects their platform during application initialization, which in turn identifies the appropriate data collection pipeline for the platform's instrument panel. A more flexible solution is envisioned for military aircraft whereby the platform may be identified from the instrument panel itself, thus automatically identifying the appropriate data collection pipeline and available flight maneuver pages. The diversity of instrument panel setups in general aviation aircraft, even within model groups, precludes this capability.

## V. Results

The following are the results of a single flight in a Garmin G1000 equipped Cessna 172S at Livermore Municipal Airport. The flight consisted of takeoff, followed by three steep turns, two Chandelles, and landing. Ground truth data was obtained using the Garmin G1000 data logging feature.

### A. Data Validation

The Aura system, mounted inside the cabin on the roof facing the left display as seen in Fig. 11, is used to transcribe the airspeed, heading, and altitude from the primary flight display. The measurements obtained from the Aura system are compared with the ground truth data shown in Fig. 12. At the beginning of data collection, the median of the first 40 non-default measurements are used to establish baselines for each flight indicator. The first valid measurement is determined to be at the index where all three flight indicators pass the rate-threshold filter as described in section III.B. The gray bar shown at the beginning of each plot indicates the period of invalid measurements before this first valid measurement. Visually, all three flight indicators appear to be very accurate.

To compute a quantitative measurement of accuracy, the ground truth measurements are interpolated to the same grid as the Aura system data. Absolute errors for altitude, airspeed, and heading are show in Table I, where measurements before

This article has been accepted for publication in IEEE Intelligent Systems. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/MIS.2025.3607236

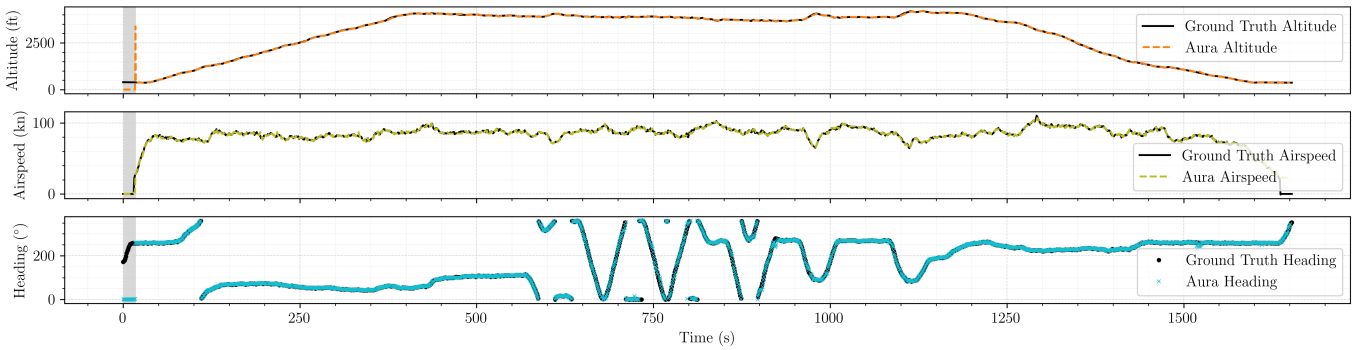



Fig. 12: Time series data of altitude, airspeed, and heading obtained from the Aura system–labeled as 'Filtered'–and the Garmin G1000 data–labeled as 'Reference'.

| Indicator | Mean Absolute Error | Median Absolute Error | Max Absolute Error |
|---|---|---|---|
| Altitude | 7.3 ft | 5.4 ft | 63.8 ft |
| Airspeed | 0.6 kn | 0.5 kn | 7.7 kn |
| Heading | 1.3° | 0.4° | 38.4° |

TABLE I: Absolute error statistics of measured flight indicators for experiment in a Cessna 172S.

| Maneuver | Precision | Recall | F1-Score |
|---|---|---|---|
| Takeoff | 1.0 | 0.95 | 0.97 |
| Steep Turn | 1.0 | 0.80 | 0.90 |
| Chandelle | 1.0 | 0.71 | 0.83 |
| Landing | 1.0 | 0.93 | 1.0 |

TABLE II: Precision, recall, and resulting F1-score from the flight data visualized in Fig. 13.

the first valid measurement indicated visually by the gray boxes in Fig. 12 are excluded. Given the relevant scales for each of these flight indicators, the median absolute errors show that the Aura system is indeed highly accurate.

### B. In-Flight Maneuver Classification

Using the in-flight measurements obtained from the Aura system, the flight indicator data can be passed through the transformer-based classifier network in the same way as the on-the-fly labeling test performed in Section III.D. The results of this experiment are presented in Fig. 13, which shows the Aura-labeled time series of the altitude with a comparison to the ground truth labeling, and Fig. 14, which shows the Aura-labeled time series of the latitude and longitude. The raw sequence classification is lightly post-processed by filling in short gaps of low confidence between regions of high confidence of the same maneuver with the high confidence maneuver label. In this context, high confidence is any classification equal or above 80% while low confidence is anything below; short gaps are defined as anything less than 30 seconds. Finally, any isolated section of the same maneuver label less than 20 seconds long is discarded.

The quantitative performance of the model is evaluated using the $F_1$ score, which is appropriate for sequence-based labeling tasks. It is defined as the harmonic mean of precision ($P$) and recall ($R$):

$$F_1 = 2 \cdot \frac{P \cdot R}{P + R}.$$

For a given label, precision and recall are defined as:

$$P = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}}, \quad R = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}}$$

where TP denotes the number of true positives, FP the number of false positives, and FN the number of false negatives. Precision measures the proportion of predicted instances of the label that are correct, while recall measures the proportion of actual instances of the label that are correctly identified. Results in Table II show that each maneuver has perfect precision, meaning there are no false positives. The recall reveals there are somewhat significant false negatives, especially for the steep turns and chandelles. Overall, the $F_1$ scores show good performance with none being lower than 0.8.

## VI. Discussion

The Aura system demonstrates that it is able to accurately transcribe flight indicator information for the airspeed, heading, and altitude while in-flight and during maneuvers. Further, the measurement is accurate enough to pass through the transformer-based classifier to obtain accurate labels for flight segments. This is particularly impressive when considering that part of the architecture of the classifier involves computing the time derivative, which is highly sensitive to noise in the original signal.

The Aura system also demonstrates robustness in its ability to continue collecting data during and after disruptions. For example, in the period between the two Chandelle maneuvers shown in Fig. 13 the camera was moved out of alignment due to one of the passengers moving in the cabin. The camera was eventually repositioned; remarkably, there is no clear indication of an increase in measurement error during or after this event as shown in Fig. 12. This is thanks to robust CV algorithms and measurement filtering with retro-active updates.

This article has been accepted for publication in IEEE Intelligent Systems. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/MIS.2025.3607236

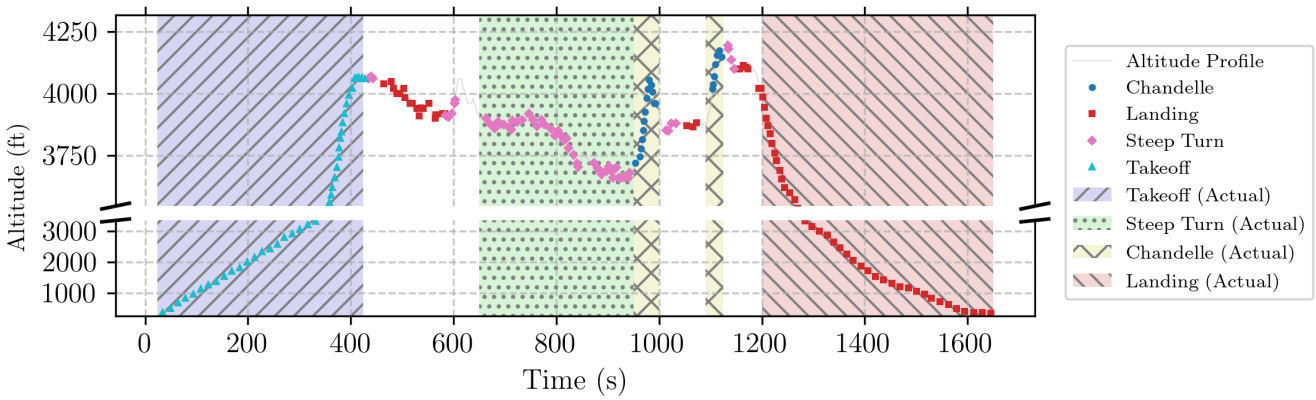



Fig. 13: Time series of the altitude with labels (markers on the altitude profile curve) obtained from the Aura system and ground truth labels (textured rectangular regions).

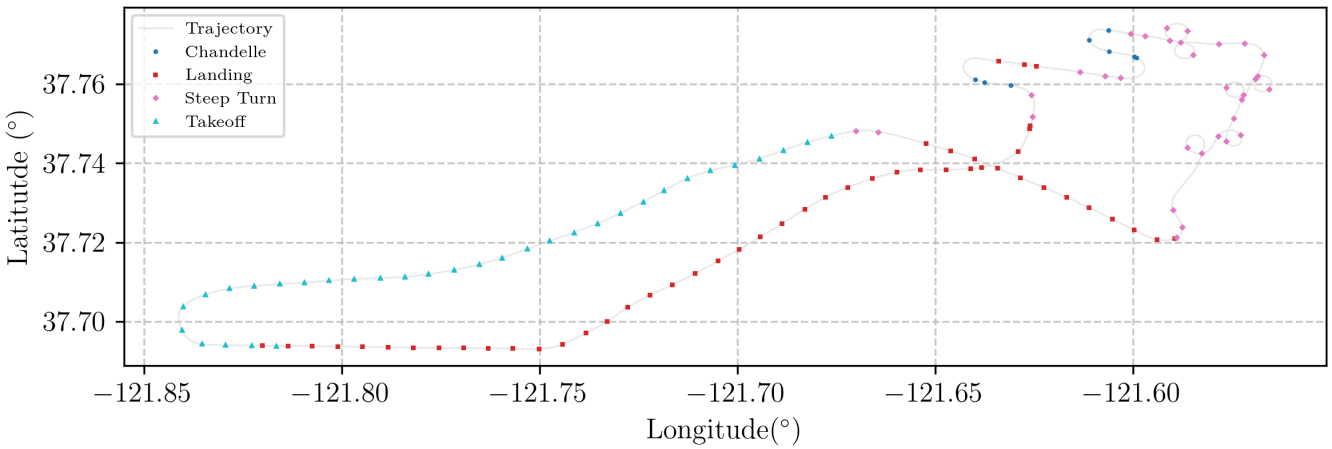



Fig. 14: Time series of the latitude and longitude with labels (markers on the altitude profile curve) obtained from the Aura system.

Considering maneuver classification, in both the synthetic testing dataset and the real Cessna 172S experiment, the transformer-based classifier is able to nearly perfectly predict the label of the labeled sections of the synthetic test dataset or the real experimental data, achieving perfect precision for the experimental data as shown in Table II. Recall is more challenging for the model, but the $F_1$ scores are at least greater than 0.8 and the overall macro average $F_1$ score is 0.92, indicating good performance. One notable existing limitation of the classifier is that it has difficulty identifying where consecutive maneuvers of the same type end and begin. This can be seen most clearly seen in Fig. 7 in the first and third row where three steep turns are executed consecutively. The classifier identifies them all as steep turns accurately, but it does not separate them. The classifier also extends the label of takeoff and landing far beyond the labeled section in the synthetic dataset. This indicates that the model cannot yet meaningfully distinguish the end of the takeoff, for example, so further work is needed to help identify these boundaries.

## VII. Conclusions and Future Work

A full pipeline, from data acquisition to pilot feedback is demonstrated using both pre-recorded video from digital and physical flight simulators as well as live footage from inside a fixed-wing aircraft in flight. The pipeline features automated flight data capture achieving median absolute error of 0.4°, 0.5 kn, and 5.4 ft for heading, airspeed, and altitude, respectively. It also demonstrates highly accurate on-the-fly maneuver labeling using the same measurements obtained from the Aura system, achieving a macro $F_1$ score of 0.92.

Future work will focus on automated recognition of flight displays and airspeed indicators. Additional flight testing in aircraft with both digital and analog instrumentation will help expand Aura's use to a variety of fixed-wing and rotary aircraft. Flight testing in different lighting conditions - dusk, dawn, high noon, and nighttime - will help ensure Aura can

operate as expected regardless of meteorological condition and takeoff time.

*All views expressed are those of the authors alone, and do not reflect the views of the US Army or Department of Defense.*

## REFERENCES


[1] J. Tai, Y. Qian, Z. Song, X. Li, Z. Qu, and C. Yang, "Research on flight training optimization with instrument failure based on eye movement data," *Journal of Eye Movement Research*, vol. 18, no. 3, p. 19, 2025.

[2] A. Ghaderi and F. Saghafi, "Enhancing pilot vigilance assessment: the role of flight data and continuous performance test in detecting random attention loss in short ifr flights," *Journal of Air Transport Management*, vol. 120, p. 102673, 2024.

[3] S. Yang, K. Yu, T. Lammers, and F. Chen, "Artificial intelligence in pilot training and education–towards a machine learning aided instructor assistant for flight simulators," in *International conference on human-computer interaction*. Springer, 2021, pp. 581–587.

[4] D. Harp, J. Ott, J. Alora, and D. Asmar, "A data-based architecture for flight test without test points," *arXiv preprint arXiv:2506.02315*, 2025.

[5] M. Jones, M. Alexander, M. Höfinger, M. Barnett, P. Comeau, and A. Gubbels, "In-flight test campaign to validate pio detection and assessment tools," *Aerospace*, vol. 7, no. 9, p. 136, 2020.

[6] B. M. de Silva, J. Callaham, J. Jonker, N. Goebel, J. Klemisch, D. McDonald, N. Hicks, J. Nathan Kutz, S. L. Brunton, and A. Y. Aravkin, "Hybrid learning approach to sensor fault detection with flight test data," *AIAA Journal*, vol. 59, no. 9, pp. 3490–3503, 2021.

[7] I. Melnyk, A. Banerjee, B. Matthews, and N. Oza, "Semi-markov switching vector autoregressive model-based anomaly detection in aviation systems," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2016, pp. 1065–1074.

[8] M. R. Schlichting, V. Rasmussen, H. Alazzeh, H. Liu, K. Jafari, A. F. Hardy, D. M. Asmar, and M. J. Kochenderfer, "Leraat: Llm-enabled real-time aviation advisory tool," *arXiv preprint arXiv:2503.16477*, 2025.

[9] W. Zhao, L. Li, S. Alam, and Y. Wang, "An incremental clustering method for anomaly detection in flight data," *Transportation Research Part C: Emerging Technologies*, vol. 132, p. 103406, 2021.

[10] P. C. Berri, M. D. Dalla Vedova, and L. Mainini, "Computational framework for real-time diagnostics and prognostics of aircraft actuation systems," *Computers in Industry*, vol. 132, p. 103523, 2021.

[11] I. Kabashkin and V. Perekrestov, "Ecosystem of aviation maintenance: transition from aircraft health monitoring to health management based on iot and ai synergy," *Applied Sciences*, vol. 14, no. 11, p. 4394, 2024.

[12] I. Stanton, K. Munir, A. Ikram, and M. El-Bakry, "Predictive maintenance analytics and implementation for aircraft: Challenges and opportunities," *Systems Engineering*, vol. 26, no. 2, pp. 216–237, 2023.

[13] A. Idowu, "Evaluating human factors in the commercial pilot-airplane airman certification standards," *International Journal of Aviation Research*, vol. 14, no. 1, 2022.

[14] P. Sorokowski and M. Garland, "Envelope expansion lessons learned," Tech. Rep., 2018.

[15] M. Smaili, J. Breeman, T. J. Lombaerts, J. Mulder, Q. Chu, and O. Stroosma, "Intelligent flight control systems evaluation for loss-of-control recovery and prevention," *Journal of Guidance, Control, and Dynamics*, vol. 40, no. 4, pp. 890–904, 2017.

[16] Y. Liu and J. Bai, "Research on quick access recorder data preprocessing based on kernel extreme learning machine and wavelet transform," in *Third International Conference on Electronic Information Engineering and Data Processing (EIEDP 2024)*, vol. 13184. SPIE, 2024, pp. 894–903.

[17] D. Johnson and T. Halbert, "Plug and play acquisition (implementing mosa)," Ph.D. dissertation, Acquisition Research Program, 2025.

[18] M. Al-Husseini, J. Barnett, A. Chen, and J. D. Thomas, "Collection, processing, and output of flight information method, system, and apparatus," Nov. 5 2024, uS Patent 12,136,277.

[19] N. Chen, Y. Man, and Y. Sun, "Abnormal cockpit pilot driving behavior detection using yolov4 fused attention mechanism," *Electronics*, vol. 11, no. 16, p. 2538, 2022.

[20] C. Qian, Z. Wang, and S. Fu, "Research on rgb-d-based pilot hand detection in complex cockpit environment," in *International conference on human-computer interaction*. Springer, 2023, pp. 573–584.

[21] E. Tunca, H. Saribas, H. Kafali, and S. Kahvecioglu, "Determining the pointer positions of aircraft analog indicators using deep learning," *Aircraft Engineering and Aerospace Technology*, vol. 94, no. 3, pp. 372–379, 2022.

[22] A. A. Gadgil and J. L. Komer, "Obstacle avoidance system," Nov. 9 2023, uS Patent App. US18/102,117.

[23] M. Al-Husseini, "System and method for calculation and display of formation flight information on augmented reality display device," Nov. 9 2023, uS Patent App. US18/102,117.

[24] J. Travert, "Flight Regime and Maneuver Recognition for Complex Maneuvers," Ph.D. dissertation, Embry-Riddle Aeronautical University - Daytona Beach, Daytona Beach, Florida, 2009.

[25] Y. Wang, J. Dong, X. Liu, and L. Zhang, "Identification and standardization of maneuvers based upon operational flight data," *Chinese Journal of Aeronautics*, vol. 28, no. 1, pp. 133–140, 2015.

[26] Barndt, Gene, Miller, Charles, Sarkar, and Subhasis, "Maneuver regime recognition development and verification for h-60 structural monitoring," 2007.

[27] C. Bodin, "Automatic Flight Maneuver Identification Using Machine Learning Methods," Ph.D. dissertation, Linkoping University.

[28] A. Kemp, "Evaluation modeling for energy management in general aviation airplanes," Ph.D. dissertation, Purdue University, 2023.

[29] S. Zhang, Z. Huo, Y. Sun, F. Li, and B. Jia, "Pilot maneuvering performance analysis and evaluation with deep learning," *International Journal of Aerospace Engineering*, vol. 2023, no. 1, p. 6452129, 2023.

[30] G. Bradski, "The OpenCV Library," *Dr. Dobb's Journal of Software Tools*, 2000.

[31] E. Olson, "Apriltag: A robust and flexible visual fiducial system," in *2011 IEEE International Conference on Robotics and Automation*, 2011, pp. 3400–3407.

[32] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, "Yolov4: Optimal speed and accuracy of object detection," 2020.

[33] D. Pal, A. Alladi, Y. Pothireddy, and G. Koilpillai, "Cockpit display graphics symbol detection for software verification using deep learning," in *2020 International Conference on Data Science and Engineering (ICDSE)*. IEEE, 2020, pp. 1–5.

[34] Y. Netzer, T. Wang, A. Coates, A. Bissacco, B. Wu, A. Y. Ng *et al.*, "Reading digits in natural images with unsupervised feature learning," in *NIPS workshop on deep learning and unsupervised feature learning*, vol. 2011, no. 2. Granada, 2011, p. 4.

[35] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[36] F. Juretić, D. Gerhardinger, A. Domitrović, and J. Ivošević, "Small piston engine aircraft vibration measurement and analysis," in *2020 43rd International Convention on Information, Communication and Electronic Technology (MIPRO)*, 2020, pp. 1705–1710.


## AFFILIATIONS


**Mahdi Al-Husseini**, Stanford Intelligent Systems Laboratory, Department of Aeronautics & Astronautics, Stanford University, California. United States Army. E-Mail: mah9@stanford.edu.

**Joshua Barnett**, Cadence Design Systems, Inc., San Jose, California. E-mail: joshuab8475@gmail.com

**Joseph Divyan Thomas**, Solidigm, Rancho Cordova, California. E-mail: josephzthomas95@gmail.com

**Tony G. Chen**, George W. Woodruff School of Mechanical Engineering, Georgia Institute of Technology, Georgia. E-Mail: tonygchen@gatech.edu.