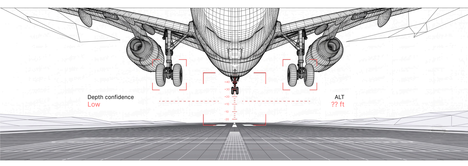
Case Study - Avionics

WHY A CAMERA-BASED LANDING SYSTEM PASSES EVERY TEST

and Still Can't Be Certified to Fly

Download



A camera-based landing guidance system uses a neural network to process a forward-facing camera feed, identify the runway, and compute the aircraft's position on final approach. In testing it works impressively, locating the runway with high accuracy under the conditions it was trained on.

Then it has to be certified, and the accuracy number that would settle the question in most industries settles almost nothing. This is the wall machine learning hits in aviation, and it is worth understanding precisely because bureaucracy is not what raises it. It is built from exactly the property that makes these systems untrustworthy in the field.

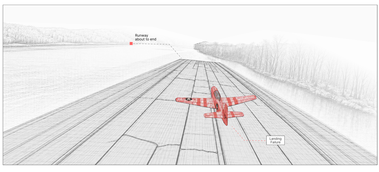
WHY AVIATION CERTIFICATION DOES NOT ACCEPT AN ACCURACY SCORE

Safety-critical airborne software has been certified for decades under a framework, DO-178C for software, DO-254 for hardware, ARP4754B at the system level, that assigns each function a Design Assurance Level according to how catastrophic its failure would be. The principle underneath that framework is simple and absolute: every requirement must trace to verifiable behavior. You specify what the system must do, you implement exactly that, and you demonstrate, line by line, that the implementation does it and nothing else. If you cannot prove it, you cannot fly it.

A neural network does not fit this model, and the reason is structural. Its behavior is not specified and then implemented; it is learned from data. There is no line of code that states the rule for recognizing a runway, no requirement that traces to a verifiable instruction because the behavior lives in millions of weights that were fit to examples. The traditional question, does the implementation match the specification, has no answer because there is no specification in the form the framework expects. An accuracy figure on a test set does not fill that gap. It reports how the system performed on the cases it was shown, which is precisely the kind of performance-based evidence the framework was built to distrust.

The aviation authorities recognized this, and rather than forcing neural networks into a model that does not fit them, EASA developed a new assurance approach, the W-shaped learning assurance process, worked out in the published CoDANN studies, specifically for machine learning. What is striking, for the purposes of this case study, is what that process demands. It requires demonstrating the bounds of the model's competence: generalization bounds that hold on unseen data, uncertainty bounds for out-of-distribution inputs, explicit identification of the edge and corner cases of the application, and verified robustness when the input drifts away from what the model was trained on.

In other words, the certification authorities are asking the one question an accuracy score cannot answer: not how often is the model right, but where does its competence end, and what happens when an input falls outside it.



WHY THIS IS EXACTLY WHERE THE MODEL IS BLIND

Here is the difficulty that makes certification more than a paperwork exercise. The visual landing system was trained on a finite set of approaches, particular runways, lighting, weather, and camera conditions. On final approach in the field, it will meet conditions that set never fully covered: an unusual runway marking, low sun directly in the lens, snow partially obscuring the threshold, haze that flattens contrast. These are not exotic. They are the ordinary variation of real flying, and they are exactly the edge and corner cases the learning assurance process requires an applicant to characterize.

When the model meets one of those conditions, it does not signal that it has left familiar territory. It resolves the unfamiliar image to the nearest representation it knows and returns a runway position with the same form, and often the same confidence, as it returns on a clear day.

> A confident, correct fix on a known approach and a confident, wrong fix on an approach the model never adequately learned are identical at the output.

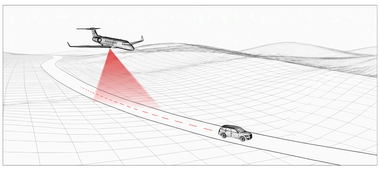
The model has no internal sense that one of them was a guess. This is the property certification is designed to catch, and it is the property an accuracy number conceals because the accuracy number is an average over conditions that were, by construction, in the training distribution.

So the applicant is caught in a genuine bind. To certify the system, they must demonstrate where its reasoning is reliable and where it is not, and show that out-of-distribution conditions are detected and handled. The model, examined only through its outputs, offers no way to do that. Its competence boundary is real but invisible.

TURNING THE COMPETENCE BOUNDARY INTO CERTIFICATION EVIDENCE

This is the gap Squint Vision Studio is built to address, and the fit with the certification requirement is close to exact. The learning assurance process asks an applicant to characterize the model's competence and detect out-of-distribution operation. That is a description of what Squint does, and the value here goes beyond producing the safeguard: it produces the safeguard in the form the authority is asking for.

Squint's Discovery tools map the model's decision-making space and distinguish the regions where the model's reasoning is well supported, the trusted regions, from the regions where inputs are sparse, unfamiliar, or ambiguous and reliability falls away. For the landing system, this map is far more than a diagnostic: it is the substance of the evidence the W-shaped process demands, translated item by item into what an applicant must submit: the trusted regions delineate the generalization bounds on unseen approaches; the ambiguous regions are the explicit edge and corner cases the process requires identified; and the boundary between them is the characterized envelope against which out-of-distribution operation is defined. Where the applicant previously had an accuracy figure and no way to answer the authority's actual question, the decision-space map answers it directly, and does so by examining the model's reasoning rather than inferring it from test performance, which is the kind of evidence the framework was built to accept.



A RUNTIME MEANS OF COMPLIANCE, NOT A GENERIC ALERT

The same analysis becomes the in-flight safeguard the assurance process increasingly expects as a means of compliance. From the decision-space map, the team builds a watchdog, a module that recognizes, in flight, when the current image has fallen into one of the model's ambiguous regions, and it is specified against the very boundary the certification evidence characterized, so that the thing monitored at runtime is the same envelope documented in the submission. This matters for certification specifically: the runtime monitor and the offline evidence are one characterization used twice rather than two disconnected artefacts, which is exactly the traceability the framework requires between what was claimed and what is enforced.

When the system meets the low sun or the obscured threshold it cannot handle, the failure is no longer silent. The watchdog flags, in the moment, that the guidance has left its characterized envelope, and control reverts to the pilot rather than to a confident wrong number. Reversion to the human when a model exits its valid conditions is a recognized means of compliance in aviation ML certification, a watchdog grounded in the model's own decision space is a direct and auditable way to implement it, because the condition that triggers reversion is the same documented boundary the authority reviewed.

WHAT THIS CHANGES FOR THE APPLICANT

Without this, the applicant is offering the authority an accuracy score against a requirement the accuracy score does not address, and the certification stalls because no one can show where the model stops being trustworthy. The number proves the model is usually right on the conditions it was tested against. Certification is asking about the conditions it was not.

Seeing where the model's competence ends converts that unanswerable question into evidence. The applicant can characterize the reliable envelope as a submitted artefact, show that departures from it are detected by a monitor specified against that same envelope, and demonstrate that the system reverts to safe control when its own reasoning is no longer sound. That is the difference between a model with an impressive test result and a system that can be trusted to fly, and in aviation, only the second one is allowed to.

The scenario described here is representative, grounded in published aviation guidance on machine-learning assurance, including EASA's CoDANN work and the W-shaped learning assurance process, the certification frameworks, Design Assurance Levels, and assurance requirements reflect the state of the field. Squint Vision Studio's capabilities, decision-space mapping, and exportable watchdog modules are described as designed.