TECHNOLOGY CONCEPT OF AN AUTOMATED SYSTEM FOR INTEGRATION TESTING

David Frisini, david.frisini@txtgroup.com, TXT e-Solutions (Italy)
Vivien-Sophie Stotz, vivien.stotz@txtgroup.com, TXT e-Solutions (Italy)
Giovanni Morlacchi, giovanni.morlacchi@txtgroup.com, TXT e-Solutions (Italy)
Vincenzo Taumaturgo, vincenzo.taumaturgo@leonardo.com, Leonardo Helicopter Division (Italy)

ABSTRACT

Looking at the trend seen in the aerospace for Verification & Validation (V&V) process, the future of integration testing will move to a more and more automated process thanks to the use of Artificial Intelligence (AI) and robotics. For the integration testing of an avionic system many different scenarios need to be executed and validated by the user running the test. Focusing on the state of the art of the integration testing, there are some issues and inefficiencies while executing the tests. Firstly, the duration of the test sets of safety-critical systems is long and involves very repetitive tasks for the operator. During this time, the system is under test and the laboratory cannot be used for other purposes which decreases the asset's availability for other stakeholders. Moreover, with distributed offices, the on-site testing could be limitating when the execution of test procedures requires at least one person physically in the laboratory. In order to mitigate these issues ARTO (Automated Robotics for Testing Optimization) has been designed. ARTO is an automated testing system with the capability of executing functional tests that up until now are being performed by test engineers, operators or pilots. It has the objective to automate the repetitive procedures in which human interactions are required, to increase the efficiency, reduce costs and make work remotely from different locations possible. The test sequence is implemented through a dedicated HMI (human-machine interface) and executed by an automated framework, fully able to carry out the tasks needed for each sequence. The system itself consists of four main subsystems: robotics, image & audio processing, framework and user interface. ARTO is designed to be applied in the integration testing field, designed to interact, and operate with the existing testing environment of the Next-Gen Civil Tiltrotor-Technology Demonstrator. It is intended to be installed inside the Simulation and Integration Laboratory cockpit where it can operate on the displays, the keyboards, the knobs and the levers. With the video and audio feedback, it can execute the test procedures that were programmed beforehand and collect test results automatically.

Abbreviations

AR	Augmented Reality
ARP	Aerospace Recommended Practice
ARTO	Automated Robotics for Testing Optimization
AFD	Adaptive Flight display
CDS	Cockpit Display System
CNN	Convolotutional Neural Networks
CSPNet	Cross Stage Partial Network
CV	Computer Vision
DoG	Derivative of Gaussian
EPGDS	Electrical Power Generation and Distribution
FCN	Fully Convolution Network
FMS	Flight Management System
HMI	Human-Machine Interface
IFR	Instrument Flight Rules
IoU	Intersection over Union
KCF	Kernelized Correlation Filter
LHD	Leonardo's Helicopter Division
LRU	Line Replaceable Unit
MAP	Mean Average Precision
MOSSE	Minimum Output Sum of Squared Error
NGCTR-TD	Next-Gen Civil Tiltrotor-Technology Demonstrator
PAN	Path Aggregation Network
PDP	Power Distribution Panel
PFD	Primary Flight Display
ReLU	Rectified Linear Unit
SAE	Society of Automotive Engineers
SIFT	Scale Invariant Feature Transform
SPP	Spatial Pyramid Pooling
SSD	Single Shot Detector
TC	Test Case
UML	Unified Modeling Language
V&V	Verification and Validation
VFR	Visual Flight Rules
VR	Virtual Reality
YOLO	You Only Look Once

1. INTRODUCTION

The cockpit is one of the most important parts in the aircraft because here the pilots interact with the instruments and give control inputs as well as reading outputs on the displays. The systems of a civil aircraft cockpit are integrated regarding the functional requirements as well as design characteristics of a variety of displays, controls and design inputs [1].

Within a modern cockpit the Cockpit Display System (CDS) has a crucial role because the information shown is necessary to perform, for example, the flight route. It represents the flight instruments that were implemented in traditional aircraft before glass cockpits were introduced. When it comes to testing the CDS, one of the issues is that there is a large number of possible scenarios to be displayed and that the images and symbols vary a lot. The CDS is composed of various flight instruments, for example the Primary Flight Display (PFD), the heading indicator, the altimeter, the airspeed indicator, the navigation display. Therefore, special attention must be paid to the testing of the CDS to check whether the data is shown correctly. This procedure is done with a human in the loop, which is usually very time-consuming, and mistakes can happen easily [2].

In the last years a lot of work has been done to realize touchscreen technology in the cockpit of commercial passenger aircraft. Now, there are already several solutions available to equip aircraft with those displays. This approach offers several advantages compared to the traditional layout of the cockpit. For the pilots it is possible to operate in a more intuitive way while controlling the systems and checking their status on the same screen. Due to the increase of the interaction between several systems it becomes more important to test the avionics while they are integrated [3].

2. COCKPIT INTEGRATION TESTING AND VERIFICATION

The goal of the aerospace system engineering process is to develop a product that fulfils all the requirements and, in the end, will be certified by the relevant authorities.

The ARP4754A published by SAE Aerospace International is the standard that has to be referred to when it comes to the development and integration of aircrafts and systems. ARP4754A describes the validation of requirements as well as the verification concerning the design implementation. The guidelines are meant for systems that have failure modes which could affect the safety and have important connections to other system in an integrated environment [4].

It is vital that systems get tested not only by themselves but also together with the external environment. Especially for avionic systems a thorough test is important because, within the aircraft the safety critical systems have to meet the safety requirements addressed in international standards [2].

The verification process ensures that the system during the development stage has been "built right". It can be carried out on any engineering element that belongs to the system or has contributed to its realization. The correct architectural and design implementation is confirmed when the system integration begins the verification. A common misconception is that verification happens after integration and before validation, but often the verification activities begin already during the development and continue through the integration and usage [5].

In the verification activity, the test design and execution play a key role in term of efficiency, safety and time-to-deliver. A typical test is performed on the specific element to quantitatively verify measurable characteristics or performance while the system is exposed to controlled conditions that can either be real or simulated. Special equipment or instruments are often used to obtain accurate data that can be analysed afterwards [5].

The testing objective is to ensure that the system or item integration carries out the intended functions which is evaluated via a pass / fail criteria. When performing the verification of the intended functions, every irregularity like an incorrect performance has to be reported for later reviewing [4].

The integration testing becomes more and more relevant since the trend in the field of system design shows that the level of integration is increasing between the functions of the airplane and all the systems that are used to implement them. Integrating systems gradually leads to many advantages but the growing complexity leads to a higher possibility of errors [6]. Another advantage of this methodology is the possibility of identifying system elements errors within the fully integrated condition that have not been identified during the previous testing steps [4].

In the avionic system verification activities, a significant effort for the integration testing has to be considered. Due to the complexity of the implemented functionalities and the interactions of the elements composing the system, it is required to involve several external dynamics (like flight model, hydraulics, fuel system, etc.), to properly stimulate the system. This is often expensive, so other techniques

can be used in order to reproduce the same outputs like simulation, emulation, simplified modelling [5].

A typical application is the CDS integration testing that involves the inspection of the indications shown on the displays in different operating situations and their correct information representation. At system level the CDS exchange data with other avionic systems and sensors. To ensure that the tests correctly identify failures in the functionality, it is vital that the inputs are provided coherently. To achieve this goal, CDS integration tests are usually done within the Simulation and Integration Laboratory, i.e. an advanced test environment able to provide all the inputs required from real or simulated equipment and navigation data provided by flight simulation.

During the system life cycle, planning this kind of tests could be a very critical topic due to the availability of this asset. So, if the test plan must be repeated every time that there has been a change in the system configuration, the cost and time-to-deliver are severely affected. Some relevant case studies could be the development of a new prototype or a new type certificate or a major change in the avionics. In these cases, a relevant amount of integration test hours should be conducted in order to demonstrate the compliance affecting the program budget heavily [2][5].

3. COMPUTER VISION - INTRODUCTION AND STATE OF THE ART

Computer vision is a field of computer science generally regarded as a subfield of machine learning. The IBM (International Business Machines corporation) definition of computer vision is the following: “Computer vision is a field of artificial intelligence (AI) that enables computers and systems to derive meaningful information from digital images, videos and other visual inputs — and take actions or make recommendations based on that information. If AI enables computers to think, computer vision enables them to see, observe and understand” [7].

3.1. Feature extraction

A feature in machine learning and pattern recognition is an individual measurable property or characteristic of a phenomenon [8].

Feature extraction has been an important research field of computer vision, some well-known state of the art feature extractors significant for the purposes of this paper are [9]:

- Canny edge detector: a differentiation-based algorithm used to detect edge points.
- Harris corner detector: a gradient-based detector used to find corners.
- Contour-based detectors like DoG (Derivative of Gaussian).
- Image descriptors, for example SIFT (Scale Invariant Feature Transform).

To identify displays in an environment the current state of the art consists of a combination of Canny edge detection and Hough transforms (or contour-based algorithms). OpenCV is a popular python/C++ library that implements these algorithms. To recognize elements such as indicators on the screen the same function can be used, along with template matching, image descriptors and neural networks.

3.2. General architecture of a convolutional neural network

Convolutional neural networks generally consist of two phases: feature learning and classification. During the feature learning phase, the neural network extracts feature from the images by convolutional layers, in which filters are applied to generate feature maps, non-linearity is introduced (a well-known function used to do this is ReLU) and pooling is executed. In the classification phase different fully connected networks, usually an artificial neural network with different layers with an activation function, provide an output consisting of the probabilities of a class for the input image. The same classification models are re-used for various other tasks such as detection and segmentation, this is often done by removing the classification head and using the backbone of the model as a feature extractor [10].

3.3. Deep learning for computer vision

During the last 15 years deep learning has expanded a lot, and with the advent of Convolutional Neural Networks (CNN) tasks like image classification and object recognition have never been this easy to implement not only thanks to GPUs becoming faster and cheaper, but also to proliferation of transfer learning. Examples of state-of-the-art CNN applications are shown in table and figure 1 and are the following:

- Image classification: the process of classifying an image assigning a label to it (for instance “dog”, “cat” or “car”).
- Object detection: a branch of computer vision that specializes in recognizing objects in an image and localizing them (generally by drawing a bounding box around the object).
- Segmentation: aims to assign a class to every pixel of an image, clustering them.

Image classification	Object detection	Segmentation
ResNet	R-CNN	Mask R-CNN
EfficientNet	Faster R-CNN	SegNet
MobileNet	YOLO	

Table 1 - Different neural networks examples for each of the tasks

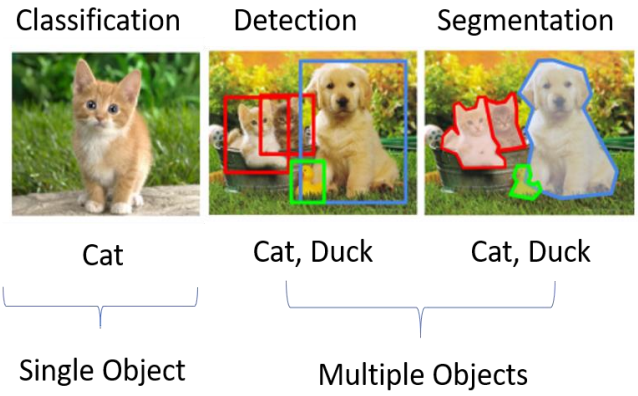

Figure 1 – Computer vision tasks [11]

Computer vision has seen an explosion in popularity with the new millennium, thanks to hardware becoming more accessible and faster (following Moore's Law) solutions that could not be scaled to real world problems are now being used over a lot of different industries and fields, from object recognition to video analysis and scene reconstruction [12]. Especially machine learning is more than ever crucial in image classification problems, with machine learning algorithms outsmarting even humans.

Examples of applications of computer vision can be found in a variety of fields. For example, insurance companies use it to analyse damaged assets, data science companies use it to recognize brands in images on social media, and the automotive industry use it to execute screen analysis, automated lane detection and road sign reading. Apart from that, computer vision is widely used in healthcare to analyse the great amount of image data produced in this field. On the topic of screen analysis in the automotive industry an example can be found where a robotic arm has been built to operate on a car's infotainment panel to automate test sequences [13].

Tan et al. [14] used 3 different state-of-the-art object detection models in the case of identifying pill and compared their performances. The three models were:

1. RetinaNet
2. Single Shot Detector (SSD)
3. YOLOv3

Algorithm	MAP
RetinaNet	82.89%
SSD	82.71%
YOLOv3	80.69%

Table 2 - Results in terms of MAP (Mean Average Precision)

Considering this they still concluded YOLOv3 would be the best choice for recognizing drug pills because of the fast detection time (achieving better than real time results, with 51 fps), shorter training time and the capability to meet the requirements of low performance platforms.

Companies are integrating computer vision systems to solve different kinds of problems, in the future, with the recent advances in the deep learning field, it will help to create precise and fast machine learning models to attain real time responsiveness.

4. OPERATIONAL ENVIRONMENT

The environment where ARTO will be validated is the LHD Next-Gen Civil Tiltrotor-Technology Demonstrator (NGCTR-TD) Avionics Full Scale Integration Rig (FSIR) environment presented herein. The NGCTR-TD provides unique capabilities in terms of software and hardware integration, testing, requirements verification and validation and simulation activities. Such capabilities have been demonstrated to be necessary for the development of a complex, highly integrated system as the Tiltrotor Electrical, Avionic and Fly By Wire (FBW) Flight Control systems.

Within this laboratory environment a real implementation of all the Electrical, Avionics and Flight Control systems of the aircraft is replicated, including the hydraulic system. This means that all the relevant real aircraft equipment for the above systems are interfaced and operated as in the real aircraft, within a representative environment and external conditions that can be simulated as required by the task or function to be tested.

The facility includes the following main areas:

- Avionics Laboratory.
- Simulation Laboratory.
- Flight Control System Laboratory.
- Electrical Laboratory.
- Hydraulic System Rigs.

Figure 2 - Cockpit mock-up structure

Figure 2 shows the overall layout of the Full Scale Integration Rig area, larger than 5000 ft^2 facility. Different colours highlight the main laboratories and the relevant rig rooms.

4.1. Avionics Laboratory

The Avionics System Laboratory provides all the software and hardware interfaces needed to perform Integration and Verification & Validation (V&V) activities related to the hardware, input/output interfaces and Software of the NGCTR-TD Avionics System.

The Avionics System Laboratory includes all the real avionics equipment of the avionics systems:

- Cockpit Display System with cursors and keyboard.
- Air Data Attitude/Heading Reference System Sensors.
- On Board computers.
- Communication System with related antennas.
- Navigation and surveillance Systems.
- Secondary power distribution units.
- Intercommunication System.
- Cockpit Voice and Data Recording System.

4.2. Simulation Laboratory

The Simulation Laboratory provides a representative human machine interface (HMI) environment, which allows the crew to operate the real commands of the aircraft, with the real control and display indications that represents a real flight scenario.

This structure contains a cockpit mock-up, fully representative of the tiltrotor cockpit, including the real controls, along with the real kinematics chain and trim actuators, in addition to a wide visual scenario (more than 180° Horizontal Field-of-View and more than ±30° Vertical Field-of-View) allowing the pilots and the operators to be completely in the loop during the flight simulated activity.

This real cockpit environment and the full out of window view including lateral sides capability allows all relevant stakeholders, including pilots, to contribute since the early stages of design and development of human interface elements,

Figure 3 - Full Scale Integration Rig Laboratories Layout

performing simulated in a representative visual scenario.

The external visual is provided by proper projectors installed above the screen dome that projects the overall flight environment view with worldwide scenario coverage sourced by real time image generator, designed and integrated taking into account the industry certification standards.

The high-fidelity image is generated by a tailored commercial scenario generator that offers several features which include VFR and IFR scenarios, weather modelling including day, night, variable conditions and navigation aids.

Real-time flight dynamics, aerodynamics data and engine parameters are provided in closed loop by a vehicle model. The Simulation Laboratory, in addition to the visual system, includes the cockpit with all its real avionics and flight control systems equipment:

- Cockpit displays including standby display.
- Cursors, keyboards and control panels.
- Autopilot control panel.
- Engine control panel.
- Pilot grips and related trims actuators.
- Fixed controls and commands.
- Landing Gear Handle.

The Electrical System Laboratory provides the system integration and testing platform for functional verification of the NGCTR-TD Electrical Power Generation and Distribution (EPGDS) design. It also allows to perform the necessary tests for certification compliance purposes.

The Electrical System Laboratory contains the real aircraft equipment of the EPGDS as:

Figure 4 - Example of simulated flight using real aircraft systems

- AC generators.
- DC starter generators.
- Generator Control Units (GCUs).
- Power Distribution Panels (PDPs).
- Aircraft battery.

The Electrical System Laboratory provides a test installation in a rig environment which is representative of the mechanical and electrical layout of the NGCTR-TD EPGDS installation on the aircraft.

By providing a fully representative rig installation it is possible to perform functional and performance EPGDS tests in a safe and controlled operating environment, thus limiting the extent of the EPGDS testing that need to be conducted on the aircraft.

The main activities performed in the electrical laboratory are:

- Functional testing of the EPGDS aimed to confirm the correct behavior of the system under normal conditions.
- Failure condition evaluation testing, aimed to confirm the correct reconfiguration of the EPGDS under abnormal conditions.
- Isolation of faults in case of abnormal condition such as short circuits or overload conditions.
- Simulation of aircraft engine start conditions.

4.3. Laboratories Modes of Operation

The development and testing of a complex system needs to be performed through different iteration stages based on increased levels of integration [15]. Every single equipment needs to be tested first as a standalone unit, then interfaced within a subsystem and finally, every subsystem needs to be connected to reach the same level of integration of the real aircraft.

As a result of this, all of the laboratories described hereinbefore can operate in several modes with different interfacing capabilities:

- Standalone testing mode as a single subsystem (Avionics, electrical).
- System Level interconnected tie-in mode, with simulation module that provides the aircraft model, flight controls model, flight dynamics, aerodynamics and navigation models, which are interfaced to the equipment or subsystem under test.
- Full Hardware In the Loop Simulation (HILS) tie-in mode, with all laboratories interfaced one each other by means of physical connections and with data shared through common interface, allowing the complete system integration tests that involve all the systems, subsystems and equipment under test.

In all these modes of operation, the various test systems allow repeatable testing through both manual and automatic test procedures. The latter is defined by the users, in accordance with the coverage of the requirements and the performance to be verified.

Moreover, the thorough monitoring and archiving of all the LRUs' interfaces data is implemented by means of dedicated hardware and software resources, providing the capability to measure the behavior of the signals identified by the user during testing activities in real time or during post-processing phases that are performed through the stored data, even after completion of the testing activities. Recorded data are available for different types of analysis involving data display in alphanumeric format or in graphic format.

All the tests can be performed by placing the real LRU within a fully simulated environment, recreating the hardware and software interfaces connected to the equipment itself, or by connecting it to the other real LRUs installed on the nearby laboratories.

The Test System allows performance of automated and manual failure injection, providing manual and automated access to individual signals for measurement or injection of failures. The Test System infrastructure is based on a data-driven architecture, where every data within each item under test is reachable for monitoring and testing purposes.

All monitored and simulated signals are mapped on a common memory and made available in real time to all the test equipment in the laboratory network, thus allowing to control the simulation and to record the testing outcomes.

5. ARTO CONCEPT DESIGN

In the last years, there has been significant development in the aerospace systems testing

sector. Despite this there are still some technological limitations that require the test engineers to be involved in repetitive and time-consuming tasks. During cockpit integration testing, the system under test cannot be used for other activities, reducing availability of the testing laboratory for other stakeholders. Moreover, with distributed offices, the on-site testing could be a limitation and removing the personnel from the laboratory can enhance operational efficiency. To solve these issues the technological concept of ARTO (Automated Robotics for Testing Optimization) has been developed.

During the development process it has been considered that the collected data and the results shall be compliant with the ARP4754A.

ARTO is designed to be integrated within the laboratory environment that is commonly used for Avionics RIGs, as described in chapter 4. It is an automated testing system with the capability of executing functional tests that up until now are being performed by test engineers, operators or pilots. ARTO's objective is to automate the repetitive procedures in which human interaction is required, in order to increase the efficiency, reduce costs, and make work from different locations remotely possible. In this way ARTO can replace the user interaction and verification done by the human.

The first configuration of ARTO focuses on test plans for the CDS within the NGCTR-TD cockpit with the objective of conducting the tests and collecting the results without an external operator. The system shall be able to recognize the AFDs and the menus needed to execute the test sequences, as well as the indicators that are involved in the tests.

ARTO consists of four subsystems which are described in detail in the following.

5.1. Robotics

The robotics subsystem is composed of a collaborative robot (cobot) that interacts with the cockpit and stimulates the system under test. It is managed by a software framework that exchanges data with the other subsystems and commands the robot in order to achieve the goal of the execution. For the proposed solution and the first configuration, the robotic arm is capable of operating on the touchscreen, push buttons and levers. The system is designed to be easily installed and removed from the cockpit, replacing the pilot seat allowing the cockpit to be reconfigured quickly for other needs or maintenance.

Before the robot can execute test procedures, the target coordinates have to be defined and mapped using xyz coordinates and stored in a database accessible to the framework. Since the robot is placed in the same position each time, the coordinates shall be fixed and checked before the test with an automatic calibration to verify the correct origin (zero) of the robot coordinates (e.g. return to home position after the abort of a test sequence, etc.). There is a target point for every button and position on the screen that is relevant for the test execution.

In order to execute the command coming from the framework, the range of the chosen robot plays an important role, especially the trade-off between the maximum and the minimum range. The maximum range allows the robot to reach buttons or displays that are far from the pilot seat, instead the minimum range, or in other words the maneuverability, is crucial since the cockpit dimension leads to some critical aspects (e.g. when a test requires an action on the control stick or aside the seat).

The trajectories of the robot have to be designed to operate also on positions that might be very close to the pilot seat or in a particular angle regarding to its own position, taking into account the geometry of the cockpit. This may vary from a helicopter/aircraft type to another and should be studied case by case, also considering the robot installation position.

A cobot is a suitable choice because its force can be adjusted precisely to match the force of a human for the actions that are usually executed by the pilot. Since in the operational environment described before there might be other personnel, the cobot was chosen due to its capability to work in an environment where humans are present, guaranteeing their safety. To satisfy the requirements for the execution of the procedures two cobots from ABB (ABB Asea Brown Boveri Ltd) are being evaluated and placed in a virtual environment including the 3D model of the cockpit of NGCTR-TD Next-Gen Civil Tiltrotor-Technology Demonstrator. In this way, it is possible to study the movement concerning the minimum and maximum range and the targets' reachability allowing to simulate different test scenarios using simulated virtual environment.

One of the aforementioned candidate is the Cobot type YuMi® - IRB 14000 which was introduced to the market in 2015. It is made to work closely together with human workers, without the need of barriers or safety-zones. It is available in two different versions, either with a single or dual-arm. Both are designed for example for part-assembly and no specialized training is required to use it. In the following the features of the single-arm cobot will be described. YuMi® is a lightweight cobot, which makes moving and installation on site easy and doable for all the workers [16].

In 2021 the cobot GoFa™ CRB 15000 was launched which is similar to the first robot described here but with a few different technical features [17].

In table 3 there is the comparison between the two candidate's robot. Both solution are designed to be interfaced with the command framework.

	YuMi® [16]	GoFa™ [17]
Weight	9.5 kg	28 kg
Payload	0.5 kg	5 kg
Max. height	835 mm	1216 mm
Base size	160 mm	173 mm
Axis	7	6

Table 3 - ABB cobot comparison

5.2. Computer Vision

The computer vision subsystem is composed of a camera and software module used to analyse information from the cockpit elements.

By joining the robotic arm with computer vision, the system will be able to recognize and operate on different instruments inside the cockpit, for example, the screens and the various indicators/buttons on them. Signal processing techniques will be employed to perform pattern and character recognition and to execute the necessary test verifications.

This capability is needed to acquire the correct visual output from the display to access information that cannot be extracted from the software but must be visualized. The scanned image will be processed by the neural network to detect patterns or text and the output can be verified in real-time. To review the test results the system will store pictures and video from the CV module and make them available to the operator. Besides that, real time image recognition is also needed in case the test sequence doesn't perform as expected, so that the visual input that caused an error to occur can be reported to the user.

The integrated software shall be connected to a microphone to verify that the audio outputs generated by the cockpit instruments are correct. This feature allows the system to verify, for instance, the crew alert messages and tones.

The development of the computer vision subsystem has been divided in 3 different phases, from the recognition of the entire screen to the single element represented on it, described in the following paragraphs.

5.2.1. Detecting screens inside the cockpit

In the first step the CV module shall be able to detect the AFDs and pass their position to the framework.

To get the screens position, the real time video is analysed in each one of its frames. These are firstly transformed into gray scale images, to which gaussian blur is applied; by doing this, sharp edges get smoothened and the noise in the image is reduced.

Screens are the brightest objects in the image, so a binary threshold is applied to the smoothened frame to segment them. To find the contours of the screens canny edge detection is applied first to the image, this ensures to get only closed contours and a more polished image, then the cv2.findContours() function is used to detect the contours of the monitors. This algorithm has shown great results over time compared to similar algorithms like Hough transforms in detecting rectangular contours [18]. The contours are sorted by area (since screens generally have the biggest area among all objects in the thresholded image) and between the 10 best candidates the polygons that could be approximated to a rectangle with an area in the expected range are chosen as the screens. Once the monitors have been recognized they can be distinguished either by their position in the cockpit or by their contents.

Finally, contours are drawn on the frame, the frame is displayed and added to the video to be saved on the internal hard drive so that the user can debug eventual errors of the video analysis. The screens can also be tracked inside the scene, different algorithms are being tested on the subject, such as MOSSE (Minimum Output Sum of Squared Error), KCF (Kernelized Correlation Filter) or Median Flow.

5.2.2. Detecting different sections of the screen content

In the second step the CV module shall be able to detect all the different sections on the screens, and to satisfy this requirement an object detection algorithm has been chosen. To train the network a dataset has been built containing labelled images of the screen sections to be detected. For the purposes of the first prototype these sections are the EICAS (Engine Indicating and Crew Alerting System), Flight Maps and the FMS (Flight Management System) menu bar.

Figure 5 - Cockpit Displays with 3 detected sections

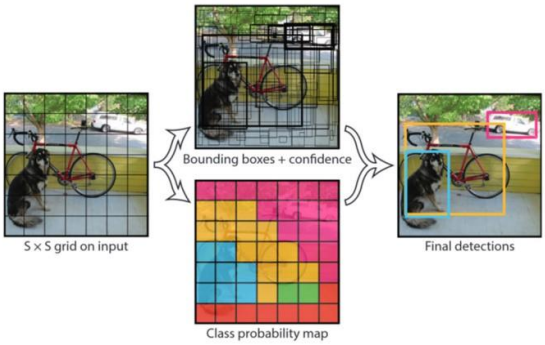

Figure 6 – YOLO's general model

The neural network algorithm chosen is YOLOv4. YOLO algorithms [19] are a family of algorithms that approaches object detection as a regression problem. This algorithm requires a single forward propagation through the neural network to detect an object, speeding up the whole process permitting real time object detection applications. YOLO works by predicting the class of the objects in the image, their bounding boxes, and the probability of the objects of belonging to the predicted class. The algorithm splits the image in a S x S grid, then each cell in the grid must predict the bounding boxes (since there could be more than an object in each cell). At last, the bounding boxes associated to a low probability of containing an object are discarded. Compared to other state of the art object detection algorithms, YOLOv4 is faster and more accurate than all available object detectors [20], it can be trained and adopted on commonly available NVIDIA GPUs (e.g., with 8-16 GB VRAM). These were the main reasons why YOLOv4 was chosen.
The network consists of [20]:

- A backbone: CSPDarknet53, a convolutional neural network and backbone that employs Darknet53 and a CSPNet (Cross Stage Partial Network) strategy to partition the feature map of the base layer into two parts and to later merge them. This allows for more gradient flow through the network.
- A Neck: SPP (Spatial Pyramid Pooling) and PAN (Path Aggregation Network).
- A Head: YOLOv3.

YOLOv4 also uses BoS (Bag of Specials) and BoF (Bag of Freebies) for both backbone and detector (an example application is data augmentation). The training process was executed on a Tesla T4 GPU during which the model obtained a MAP of 98.64% with an IoU (Intersection over Union) threshold of 50%. IoU is a popular metric for evaluating object detection models, defined by Google [21] as: ”A value used in object detection to measure the overlap of a predicted versus actual bounding box for an object. The closer the predicted bounding box values are to the actual bounding box values the greater the intersection, and the greater the IoU value”.

Tests on the actual implementation of the neural network will be executed with particular attention to the MAP value and the frames per second, in order to achieve trustworthy results and real time performances.

5.2.3.Segmenting indicators on screen

To execute the test verifications the capability to detect different indicators on the screen is needed and to do this, various feature extraction techniques are being tested. Amongst them are Canny edge detection, Harris corner detectors, template matching and deep learning approaches similar to the one described in section 5.2.2.

As an example of application, an algorithm has been implemented to detect the landing gear indicators on the screen, using a combination of Hough transforms and Canny edge detection. With this method, it is possible to detect the indicators in every frame of a video, this way their coordinates could be retrieved and the indicators tested.

5.3. User Interface

The User Interface (UI) allows the user to manage the functions of ARTO and lets the operator design, execute and collect/review results of the specific test.

The UI shall be implemented as a web application and the user shall access as a client from different devices. Two different sections and functionalities shall be displayed: the test creator and the test runner.

In the test creator section, the user shall be able to set up the logical sequence of a test, choosing from different types of test steps represented as boxes. The overall test procedure will always be shown in the lower part of the window in a logical flowchart, with

the set and verification steps. The tests shall be saved in a database and different test sets could be created.

In the test runner section, the user has the possibility to load the saved tests and execute them sequentially in a fully automated mode. The status of the test's execution and the error logs shall be retrievable at any time. The test shall be executable from any point of the flowchart and the user is also able to skip some steps by hiding them.

5.4. Framework

The framework is the software that manages the connection among all the subsystems, it establishes a common practice for creating, interpreting, analysing and using architecture descriptions within a particular domain of application [22]. Moreover, the framework shall be able to manage the external resources already present in the test RIG. The Unified Modeling Language (UML) was used to describe the architecture of the system to provide a general idea of how it was developed. Figure 7 shows ARTO's architecture.

The user employs the test creator section of the User Interface to create the test case that has to be executed or load a saved test. The test case is then initialized and the input data is transferred to the human-machine interface (HMI) Runner to start the test. Now, the information is processed by the Test Case (TC) Runner, which translates the test action into an actual command. Both the Status Cockpit structure and the test command are provided to the Robot Manager, which is responsible for the connection between the robotic arm and the TC Runner. The Status Cockpit structure must be updated after each step to keep track of the current state of the cockpit elements involved in the test procedure. It includes the logic state of buttons, and levers and the visual output shown on the AFDs. Then, the Robot executes the command and notifies the Robot Manager when the action has been performed. In the first configuration, the robot is not guided by Computer Vision in choosing the coordinates of the end point involved in the test, so it has to access to a set of predefined coordinates that identify a limited set of reachable locations. In anticipation of future development phases, the framework has been set up to manage the information passed from the CV module to the Robot Manager and make the dentification of target points autonomous according to the specific task required by the current test.

The information is transferred to the TC Runner which communicates with the Computer Vision to verify whether the actual result matches the expected one. Verifiers shall be created to manage different types of output data that comes from both internal and external resources, by doing this the information is elaborated, classified and verified. Finally, the outcome of this function is sent to the TC Runner,

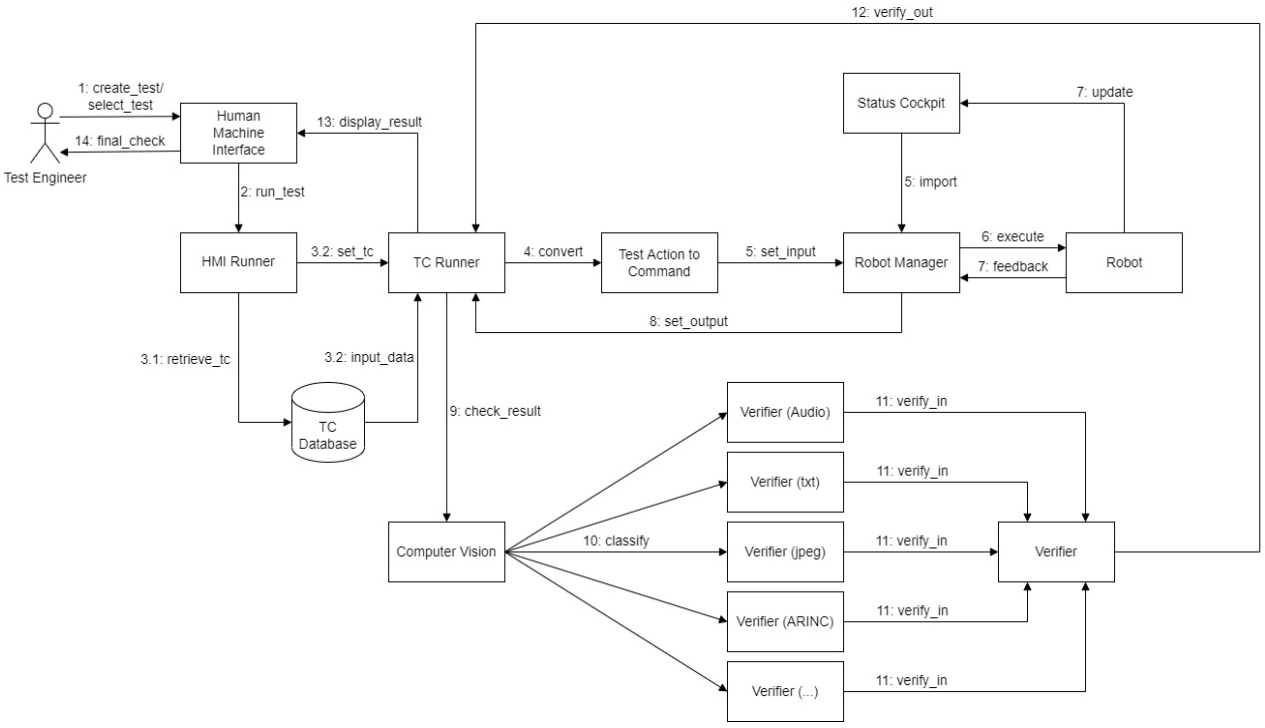

Figure 7 – ARTO collaboration diagram

which displays the final result through the HMI. A log file is created in order to keep track of each test step, as the test proceeds.

6. CONCLUSION

In comparison to other approaches that have already been studied by research institutes and companies (e.g. robotics for emergency procedures, support for distributed crew and hardware acceptance tests for panels/displays) ARTO introduces a new step in the testing industry which is the capability to create an automated verification process for complex system functionalities. The artificial intelligence allows ARTO to acquire several information from the cockpit's indications that right now are only readable by a human.

The key points that make ARTO a cost-effective solution for integration testing are:

- The integration testing phase will allow a shorter time to deliver of the test reports from the integration testing to the test engineer.
- The testing execution effort will be reduced to a minimum: the operator will not be required to interact with the system under test during the testing run.
- The enhanced testing framework is able to perform the test sets and collect all the results automatically.
- The quality of the testing activity will increase for the test engineer: the effort will focus on designing the test and not on executing it.
- The testing process will become more reliable due to the high repetitiveness and human factor being reduced.
- The testing activity could be done remotely: operators do not need to be in the cockpit.
- The high repetitiveness and the traceability of the testing procedure using ARTO, enable the user to adapt it to the aeronautical standard ARP4754A.

7. FURTHER DEVELOPMENTS

In the next years, further developments will be carried out through different configurations. The conceptual ideas for the next two configurations will be described in the following.

The second configuration will be able to perform tests on the touch display and on the cockpit panel close to the displays, therefore, the robot will be able to use several kinds of end-effectors depending on the action that has to be executed. It will include an audio analysis subsystem to conduct the tests that need audio verification. Because of this, a microphone will be integrated in the CV subsystem. The image dataset needs to be expanded with images required for neural network training and pattern matching. Text recognition must be implemented to include the verifications tied to reading messages from the cockpit, this will probably be done using transfer learning on a neural net. A library containing all CV functions must be developed as the framework will need to work with them. The User Interface will also be accessible using a tablet and it will be possible to visualize on the display the test step which is being executed thanks to the logical boxes that light up in the flowchart as the test proceeds.

The third configuration will be able to recognize a wider range of displays/panels interactions with more objects. For the movement of the robot the Computer Vision will be able to see the whole cockpit in its different sections and communicate with the framework to identify the target with the camera, so, the robot will be able to move without the use of predefined targets. The cockpit model will also be integrated and imported in the VR/AR/XR programs in order to manage the system remotely.

The technology can be further developed to fully operate in the cockpit and execute the entire flight procedures. From here on, it will be attainable to assist pilots either in the simulator while training or later even in flight, implementing the concept of distributed crew. Additional applications of this technology could be used to realize autonomous flight testing in emergency or normal conditions, reducing the risk for experimental test pilots.

BIBLIOGRAPHY

[1] F. L. Q. K. J. Zhang, “Civil Aircraft Cockpit Human Machine Interactive Dynamic Assessment Quality Improvement Based on System Engineering,” *Human-Computer Interaction. Theory, Methods and Tools,* p. 91–298, 2021.

[2] M. U. K. H. S. M. Z. Iqbal, Testing cockpit display systems of aircraft using a model-based approach, Germany: Springer-Verlag GmbH , 2020.

[3] A. M. D. Z.-. M. J. G. M. Xuereb, “Multi-modal Interaction Between Pilots and Avionic Systems On-Board Large Commercial Aircraft,” in *Engineering Psychology and Cognitive Ergonomics. Cognition and Design*, Cham, Switzerland, Springer International Publishing AG, 2017, pp. 200-210.

[4] Guidelines for Development of Civil Aircraft and Systems, ARP4754A, 2010.

[5] G. J. R. K. J. F. R. D. H. T. M. S. (. David D. Walden, System Engineering Handbook: A Guide fpr System Life Cycle Process and Activities (4th ed.), San Diego CA: International Counsil on System Engineering. Published by John Wiley Sons, Inc., 2015.

[6] Software Considerations in Airborne Systems and Equipment Certification, DO-178C, 2011.

[7] “www.ibm.com,” [Online]. Available: https://www.ibm.com/topics/computer-vision. [Accessed 05 07 2022].

[8] C. Bishop, Pattern recognition and machine learning, 2006.

[9] S. Ehab and Q. Murad, “Recent Advances in Features Extraction and Description Algorithms: A Comprehensive Survey,” *CoRR,* vol. abs/1703.06376, p. 5, 2017.

[10] “deci.ai,” [Online]. Available: https://deci.ai/blog/sota-dnns-overview/. [Accessed 05 07 2022].

[11] A. Khan, “Machine Learning in Computer Vision,” *Procedia Computer Science,* vol. 167.

[12] “medium.com,” [Online]. Available: https://medium.com/@laranguyen/what-is-computer-vision-past-present-and-future-806a7263ee08. [Accessed 05 07 2022].

[13] “www.gadgeon.com,” [Online]. Available: https://www.gadgeon.com/success-stories/cv-and-ai-based-car-infotainment-panel-test-automation/. [Accessed 05 07 2022].

[14] L. Tan, T. Huangfu, L. Wu and W. Chen, “Comparison of RetinaNet, SSD, and YOLO v3 for real-time pill identification,” *BMC Medical Informatics and Decision Making,* vol. 21, no. 1, p. 11, 2021.

[15] N.-2.-6. Rev1, Systems Engineering Handbook, NASA Headquarters, Washington, D.C. 20546: National Aeronautics and Space Administration, December 2007.

[16] “https://global.abb/group/en,” [Online]. Available: https://search.abb.com/library/Download.aspx?DocumentID=9AKK107046A3807&LanguageCode=en&DocumentPartId=&Action=Launch. [Accessed 18 07 2022].

[17] “solutions.abb/cobots,” [Online]. Available: https://assets.ctfassets.net/gt89rl895hgs/1MBowsjHDvAEykEwKKBWwl/72d6fab3961eb01387b5dfb06c36edf7/GoFa_CRB15000-datasheet_edited.pdf. [Accessed 18 07 2022].

[18] s. R. Kabir, M. Akhtaruzzaman and R. Haque, “Performance Analysis of Different Feature Detection Techniques for Modern and Old Buildings,” vol. 2280, p. 8, 2018.

[19] J. Redmon, S. Kumar Divvala, R. B. Girshick and A. Farhadi, “You Only Look Once: Unified, Real-Time Object Detection,” *CoRR,* vol. abs/1506.02640, p. 10, 2015.

[20] C.-Y. W. H.-Y. M. L. Alexey Bochkovskiy, “YOLOv4: Optimal Speed and Accuracy of Object Detection,” *CoRR,* vol. abs/2004.10934, 2020.

[21] “cloud.google.com,” [Online]. Available: https://cloud.google.com/vision/automl/object-detection/docs/evaluate. [Accessed 18 07 2022].

[22] “http://www.iso-architecture.org/,” ISO/IEC/IEEE 42010 , [Online]. Available: http://www.iso-architecture.org/42010/cm/. [Accessed 14 07 2022].

[23] J. R. I. J. Grady Booch, Unified Modeling Language User Guide, The, 2nd Edition, Addison-Wesley Professional.