Proceedings of the 2021 AIAA/IEEE 40th Digital Avionics Systems Conference (DASC), 3-7 October, San Antonio, USA
DOI:10.1109/DASC52595.2021.9594312

Autonomous Ground Refuelling Approach for Civil Aircrafts using Computer Vision and Robotics

Suleyman Yildirim
School of Aerospace,
Transport and Manufacturing
Cranfield University
Bedford MK43 0AL, UK
suleyman.yildirim@cranfield.ac.uk

Zeeshan A. Rana
School of Aerospace,
Transport and Manufacturing
Cranfield University
Bedford MK43 0AL, UK
zeeshan.rana@cranfield.ac.uk

Gilbert Tang
School of Aerospace,
Transport and Manufacturing
Cranfield University
Bedford MK43 0AL, UK
g.tang@cranfield.ac.uk

***Abstract*—3D visual servoing systems need to detect the object and its pose in order to perform. As a result accurate, fast object detection and pose estimation play a vital role. Most visual servoing methods use low-level object detection and pose estimation algorithms. However, many approaches detect objects in 2D RGB sequences for servoing, which lacks reliability when estimating the object's pose in 3D space. To cope with these problems, firstly, a joint feature extractor is employed to fuse the object's 2D RGB image and 3D point cloud data. At this point, a novel method called PosEst is proposed to exploit the correlation between 2D and 3D features. Here are the results of the custom model using test data; precision: 0,9756, recall: 0.9876, F1 Score(beta=1): 0.9815, F1 Score(beta=2): 0.9779. The method used in this study can be easily implemented to 3D grasping and 3D tracking problems to make the solutions faster and more accurate. In a period where electric vehicles and autonomous systems are gradually becoming a part of our lives, this study offers a safer, more efficient and more comfortable environment.**

***Index Terms*—autonomous, aircraft, refuelling, robotics, integration**

I. Introduction

Aircraft refuelling is accompanied by attendant hazards which must be managed sufficiently for their mitigation to acceptable levels. The issues are much the same whether the fuel source is a tanker/bowser or a fuel hydrant system. The primary risk is the unintended ignition of fuel vapour, which can occur by a single spark. A sufficient quantity of fuel vapour can create a high risk of ignition which may result from the spillage arising from procedural errors, leaks, aircraft tank venting or failure of pressurised fuel lines or their couplings. In connection with refuelling, there have been many accidents in the past that caused a great loss of life and property. To eliminate this, trained and skilled personnel are required for this operation. However, training personnel to work in a high-risk job requires both a lot of time and money. No matter how trained and skilled they are, there is always an accident risk wherever people work. To reduce the risks and increase safety and comfort in the airline industry, visual servoing systems can achieve this easily.

Aircraft refuelling is a serious task which is carried out by trained aircraft mechanics. To enable the aircraft to continue its journey, refuelling process where trained personnel carried out task to fill an aircraft with fuel needs to be done. There are two methods of refuelling; gravity refuelling and pressure refuelling [1]. Small aircraft such as Cessna 172 uses gravity refuelling, when it comes to large aircraft such as Boeing 737 pressure refuelling (Shown in Figure 1) is used. Safe refuelling operations require strict adherence to procedures and careful application of the safety precautions, not only by the refuelling operators but also flight crew, the cabin crew and the other ground operators [2].

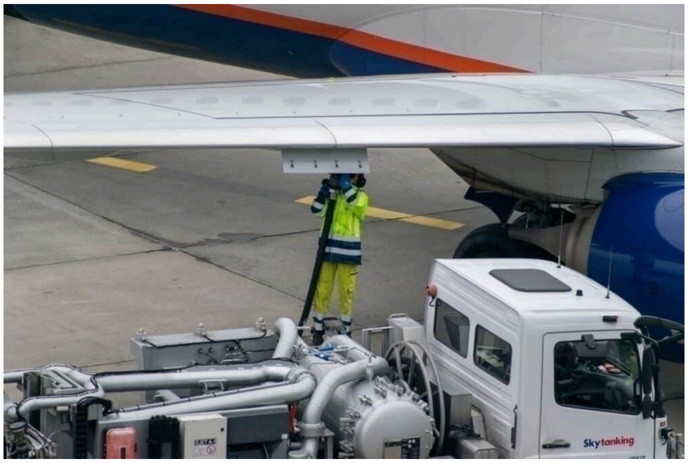

Fig. 1. Pressure aircraft refuelling

Robotic manipulators are very useful in many conditions, such as risky or unpredictable environments. Robotic manipulators that are autonomous or operated remotely can greatly reduce the number of people needed for a given task. Because of the primary risk of unintended ignition of fuel vapour caused by a single spark, it is necessary to conduct this research in order to reduce the amount of incidents and risks that can result in fatal accidents [3].

In the past, the following incidents happened [4]:

1) On 5 September 2001, a British Airways Boeing 777-200 on the ground at Denver USA, was substantially damaged, and a refuelling operative killed, when a fire broke out following the failure of a refuelling coupling under pressure because of improper attachment.

2) On 13 April 2010, a Cathay Pacific Airbus A330-300 en route from Surabaya to Hong Kong experienced difficulty in controlling engine thrust. As these problems worsened, one engine became unusable. Salt water contamination of the hydrant fuel system at Surabaya after alterations during airport construction work was found to have led to the appearance of a polymer contaminant in uplifted fuel.

3) On 16 April 2014, a pre-flight concern about whether a Boeing 777-200ER about to depart Singapore had been over-fuelled was resolved by a manual check. The Investigation found that a system fault had caused over-fuelling and that the manual check carried out to confirm the actual fuel load had failed to detect it because it had been not been performed correctly.

4) On 7 June 2016, a Boeing 777-300 made a high speed rejected takeoff on 3200 metre-long runway at Dhaka after right engine failure was enunciated. The Investigation found that engine failure had followed Super Absorbent Polymer contamination of some of the fuel nozzle valves which caused them to malfunction leading to Low Pressure Turbine mechanical damage.

A refuelling operation scenario follows:

1) The aircraft and the fuel tanker have to be grounded.
2) Before removing the filter cap, the fuelling nozzle grounding cable has to be connected to the aircraft grounding receptacle.
3) After grounding procedure, the fuel nozzle needs to be inserted carefully into filler cap and commence refuelling.
4) Stop the refuelling once the desired fuel quantity has been reached.
5) Once the refuelling process is done, install and secure the filler cap.
6) Remove the refuelling nozzle grounding cable from the aircraft grounding receptacle.
7) Remove the grounding of aircraft and the fuel tanker.
8) Ensure there is any spillage of fuel on the ground.
9) The aircraft is ready to depart.

A direct objective of this study is to develop an autonomous ground refuelling approach, which uses computer vision, machine learning and visual servoing methods in order to locate the pressurised fuel servicing adaptor and place the nozzle in slipway. Trajectory information that is identifying and locating the pressurised fuel servicing adaptor needed for the robot manipulator. The autonomous refuelling approach is going to use the visual servoing to close the loop around the motion control problem [5].

This paper introduces novel ”PosEst” method to enable the 3D operations in high accuracy and speed. Relative object detection and tracking methods are using low-level object detection and pose estimation algorithms such as arc/contour detection. Consequently, new solution for refuelling adaptor detection and pose estimation in autonomous ground refuelling operations is presented in this paper. The main contributions of this method include two aspects. In the aspect of refuelling adaptor detection, convolutional neural networks have been trained using 2D RGB dataset and 3D depth stream for faster and accurate detection to solve the problem in multi-scale. In the aspect of pose estimation, different pose estimation algorithms have been tested and implemented on the basis of the structure feature of pressurised refuelling adaptor, which takes advantage of the structure characteristics of refuelling adaptor to solve the problem. Thereby the study offers the-state-of-the-art method to detect refuelling adaptor and obtain its pose in autonomous ground refuelling approach based on the combination of 2D object detection/3D object tracking and pose estimation.

As the introduction to autonomous ground refuelling system is presented in Section 1, the rest of the paper is organised as follows: Section 2 outlines existing methods related to aircraft refuelling.The brief explanation of the proposed method can be found in Section 3. Section 4 covers the dataset preparation, machine learning model and its structure, pose estimation algorithms. The results and the discussion can be found in Section 5. Finally the conclusion is presented in Section 6.

II. Related Work

For autonomous ground refuelling system to achieve safe approach and coupling procedure and increase its robustness, accurate detection of the object and its pose are vital.

Existing visual measurement methods are mostly based on artificial features. Due to their susceptibility to occlusion, artificial features such as spray marks or LEDs caused some problems. VisNav, short for visual navigation system, developed by Valsek based on artificial features [7]–[11]. The LEDs emit with different frequencies are mounted on the system to detect the centre of the beacon with measuring units. On the receiver side which generates a current according to acquired modulated lights, position sensing diode is mounted. Gaussian least-squares differential correction algorithm is used to calculate the refuelling adaptor’s pose which is a combination of beacon data. The main advantage of this method is its ability to reduce the inference filtering the light out on specific frequency bands in short distance. On the contrary, method produces poor signal to noise ratio in long distance due to the low-energy intensity acceptance of position sensing diode.

The method which uses both vision system and global positioning system's fusion switch strategy is proposed by Pollini [12]. To identify the refuelling adaptor, near infrared filter and CCD camera are used along with mounted LEDs to object. The distance between refuelling adaptor and nozzle is measured using machine vision with installed marker points as the position between tanker and the aircraft is measured using GPS. To construct the best relation between 3D marker and 2D feature Lu, Hager and Mjolsness [13] pose estimation algorithm is used with a fixed number of steps.

Using colour analysis, contour analysis and relative position measurement algorithm Wang proposed a method to detect the refuelling adaptor coating with a material has high reflection characteristic [14], [15]. The projection of refuelling adaptor and its characteristic relationship between short and long axes can be used to obtain the yaw and the pitch angles by combining the yaw and the pitch angles with the pose measurement algorithm. In order to use the high reflection characteristic of coating material, refuelling adaptor needs to be modified in advance and forming the circle from elliptical projection must be considered.

Using 3D Flash LIDAR camera and level set front propagation method Chen is able to segment the image, identify the returned colour and depth stream from LIDAR camera and finally determine the desired object from the multiple segments [16], [17]. The 3D point cloud data is combined with RANSAC algorithm [18] to identify the position of refuelling adaptor as the pose estimation of refuelling adaptor is not applied. Moreover, modification of refuelling adaptor is needed as it is based on artificial features.

There are many studies proposed based on grey scale and the shape of the refuelling adaptor to detect the object and estimate its pose [19]–[23]. Based on the contour feature of refuelling adaptor and the threshold, Yin proposed a method [19], [20] which uses spatial relationship between inner region's elliptical projection and the shape of the inner region to detect the refuelling adaptor and its pose. The drawback of the method is being not suitable for long distance measurement as the ambiguity of circular feature projection calculation is ignored.

Song's detection strategy [21] is based on low-rank, multi scale, sparse decomposition. As the method perform the detection without any structural characteristics of the object, it is highly susceptible to illumination in terms of foggy and cloudy environment and can fail in highly complex surrounding.

Martinez's visual measurement scheme [22], [23] based on the direct methodology has four stages: initialisation, detection, tracking, and position estimation. In the detection stage two different methods are being used: image threshold segmentation and edge image template matching. According to the studies which use only characterisation of the object for detection, this method offers higher success rate. The hierarchical multi-parametric and multi-resolution implementation of the inverse compositional image alignment strategy [24] is prefered for tracking stage. The HMPMR-ICIA technique can achieve stable tracking with scale invariance. The four points on the object determined as referral points in order to use the world coordinate system to define according to diameter of the object. To calculate the relative position of the refuelling adaptor, the four referral points are transformed into homography matrix. Template matching forms the detection stage of the method and it is the most time consuming stage as it includes different variations of the object such as illumination, scale and position. Empirical threshold is applied to segment the input image to detect the object if the template matching fails as it is not achievable to gather all conditions into the template. During the position estimation stage the pitch and yaw angles are ignored, therefore the pose estimation can not be comprehended fully.

Fortunately, machine learning techniques have offered satisfactory solutions to many problems [25]–[27]. There are several studies adopted machine learning techniques to detect the refuelling adaptor. Yin proposed a support vector machine [28] performing block type classification method on the object. Another method proposed by Wang which is using convolutional neural networks [29].

III. System Design

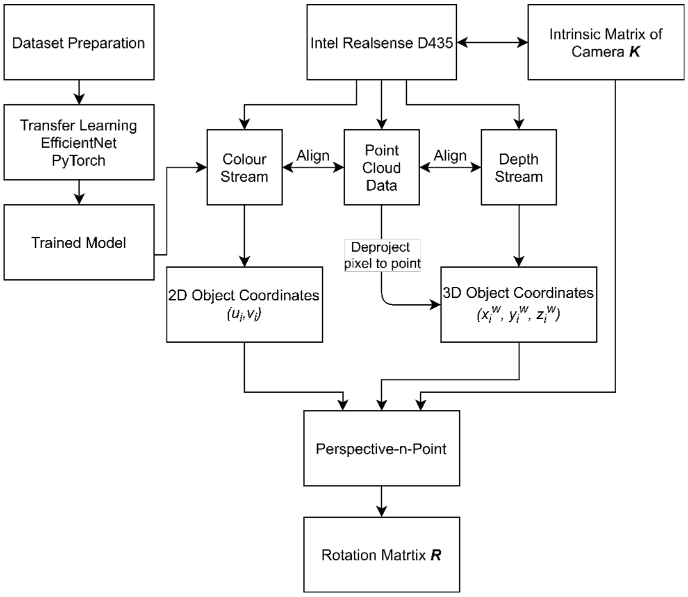

Fig. 2. PosEst workflow

A novel PosEst method (Can be seen in Figure 2) does not require any artificial features of the refuelling adaptor to detect and determine its pose. Using the custom created dataset, custom EfficienNet-B0 model has been trained on PyTorch framework. After successful detection in colour stream, the refuelling adaptor is also being detected in depth stream. The real world coordinates of refuelling adaptor can be derived using point cloud data in high accuracy. Using 2D coordinates and 3D real world coordinates, its pose can be derived using Perspective-n-Point algorithm from streams in real-time.

The proposed method offers high speed, high accuracy and easy implementation in real aviation problems. Autonomous ground refuelling operation is a key to digital aviation. According to IATA [33], it's been foreseen that advanced biometrics, autonomous robotic systems, greener energy sources, VR/AR are going to play key roles in the future of the airline industry.

A autonomous refuelling scenario as follows:

1) The aircraft needs to be in the park position.
2) The autonomous refuelling system pulls up to aircraft.
3) The manipulator swings towards aircraft, near the refuelling adaptor.
4) Using the 3D cameras and already constructed deep learning algorithms, boom finds the refuelling adaptor and guides towards to refuelling adaptor.
5) The fuel nozzle needs to be inserted carefully into refuelling adaptor and commence refuelling.
6) Stop the refuelling once the desired fuel quantity has been reached.
7) The boom is detached from the adaptor and stowed once aircraft has been refuelled.
8) The autonomous refuelling system pulls away from the aircraft.
9) The aircraft is ready to depart.

IV. Methodology

A. Dataset Preparation

Pressure refuelling adaptor [36] is a connection adaptor for the delivery of pressurised fuel to aircrafts. The design and construction of the refuelling adaptor must conform to both MIL-A-25896 and MS24484-5 standards. Military Standard, "MIL-STD", is a United States defence standard and helps to fulfil standardisation objectives by the U.S. Department of Defence. Standardisation is beneficial in achieving interoperability, commonality, total cost of ownership, reliability, ensuring products meet certain requirements, compatibility with logistics systems and defence related objectives [37]. Adaptor is generally constructed of aluminium and high-strength stainless steel to ensure the maximum durability and strength. The poppet assembly, adaptor body and spider are precision investment cast for a long-service life under high usage conditions.

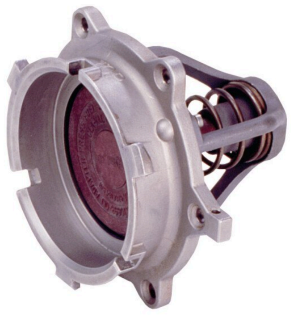

Fig. 3. Pressurised fuel servicing adaptor [6]

Custom dataset plays fundamental and important role due to the lack of acknowledged dataset for autonomous ground refuelling as successful detection is considered. To be able to train robust deep learning model, refuelling adaptor dataset is collected from Boeing 737-400 aircraft. To make sure to custom trained model works under any conditions, different augmentation methods (Shown in Figure 4) such as brightness, exposure, blur, flip have been applied to dataset. The dataset contains 567 training, 162 validation and 81 test images.

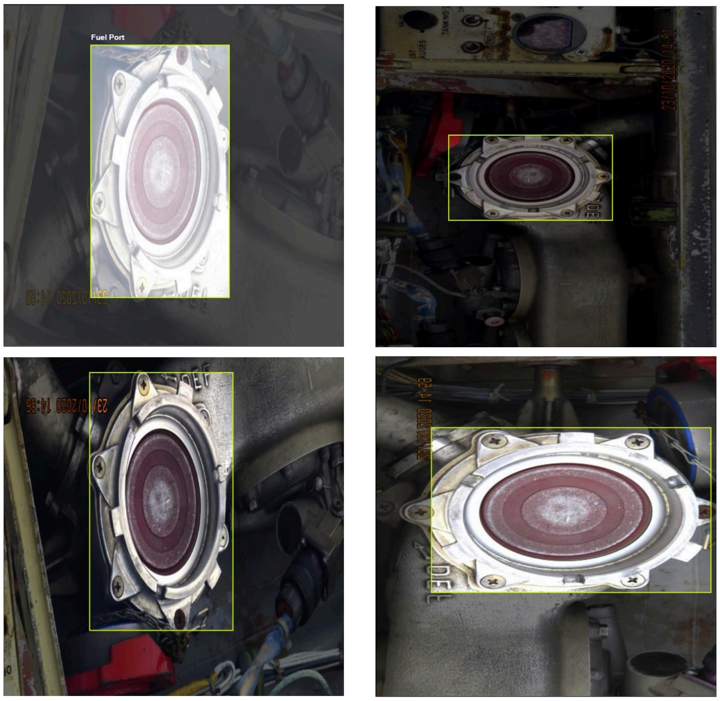

Fig. 4. Labelled dataset

The dataset has been collected using Intel® RealSense™ D435 [35] (Shown in Figure 5) depth camera. The camera has Intel® RealSense™ D4 Vision Processor and it can stream both RGB up to 1920x1080 resolution and Active Stereo Depth up to 1280x720 resolution. Dual global shutter sensors stream up to 90 FPS and their Field of View is over 90°. It can accurately range from 20 centimetres to 10 metres [34].

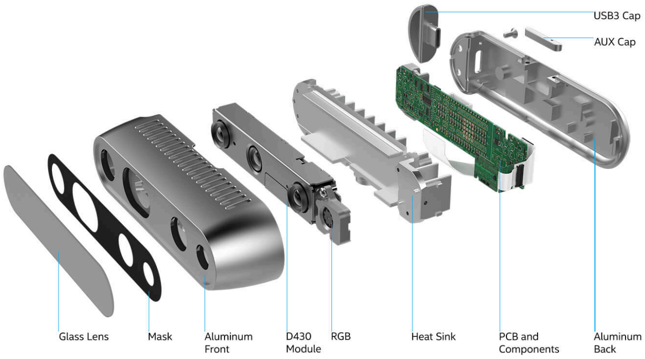

Fig. 5. Intel® RealSense™ D435

B. Network Architecture

When more computational power is available, convolutional neural networks which are commonly developed at a fixed resource cost, scale up to achieve better accuracy. For instance, only by increasing the number of layers ResNet can be scaled up from ResNet-18 to ResNet-200 [38]. The conventional approach for model scaling is to use larger input image resolution for training or to arbitrarily increase the width or depth of layers. Even these offer better accuracy, often yield sub-optimal performance and need tiring manual tuning. To obtain better convolutional neural network in efficiency and accuracy, different kind of method has been adopted as a principle [32].

EfficientNet [30] is a convolutional neural network as well as being a scaling method that scales all dimensions of width, depth, and resolution uniformly using compound coefficient. EfficientNet's scaling approach scales network's depth, width and resolution uniformly with a set of fixed scaling coefficients unlike the conventional practice which is arbitrary scaling the factors. To use 2^N times more compute power, constant coefficients α, β, γ determined by grid search on the original model where they are used as the increase of the depth by α^N, the width by β^N and the image size by γ^N. Instead of using different number of coefficients, EfficientNet employs a compound coefficient ϕ to scale network uniformly [39]. As the convolutional neural network is going to need more layers to capture fine-grained patterns as the input image gets bigger, balancing all the dimensions of the network gives better overall performance on the contrary scaling depth, width and resolution by different coefficients. Addition to MobileNetV2's squeeze-and-excitation blocks, EfficientNet-B0 network also uses the inverted bottleneck residual blocks of MobileNetV2 [40].

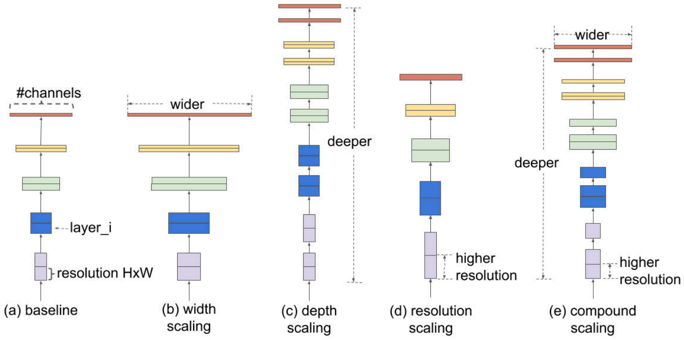

Fig. 6. Comparison of scaling methods [32]

Baseline network also affects heavily the effectiveness of the model scaling. To increase the performance of the model further, new baseline network has been developed using AutoML MNAS framework [41] which optimises both efficiency in terms of FLOPS and accuracy. The newly developed baseline network is similar to MnasNet and MobileNetV2 as it is using mobile inverted bottleneck convolution but it is slightly different due to the increase in FLOPS.

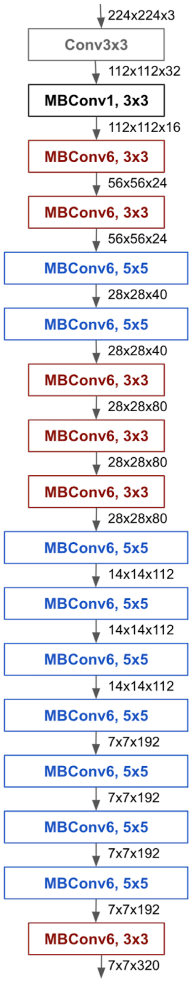

Fig. 7. The architecture of EfficientNet-B0 [32]

In contrast to conventional scaling methods, scaling up baseline models such as ResNet and MobileNet using compound scaling method increases model's efficiency and accuracy consistently. While reducing both FLOPS and parameter size, EfficientNet achieve better efficiency and higher accuracy over existing convolutional neural networks on ImageNet [42] dataset. It can be seen in both figure 8 and figure 9.

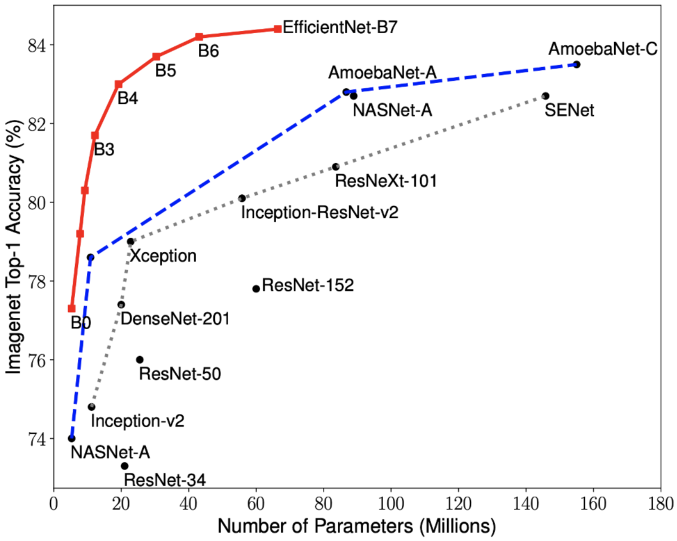

Fig. 8. Accuracy vs. Model Size [31]

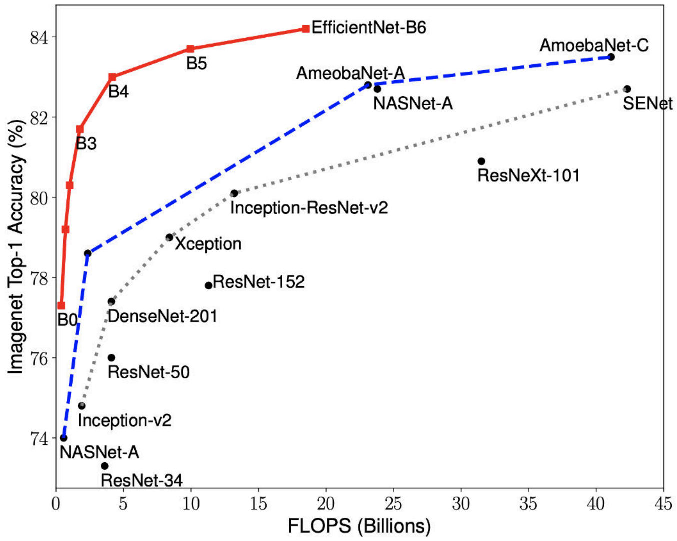

Fig. 9. Accuracy vs. FLOPS [31]

To reveal the real performance of EfficientNet, other datasets should be transferred even it is performing well on ImageNet. To test this, well known datasets such as Flowers [43] and CIFAR-100 [44] has been used. Even with an order of magnitude reduction, EfficientNet obtained 98.8% and 91.7% accuracies on Flowers and CIFAR-100 datasets respectively. With the significant improvements in both accuracy and efficiency, EfficientNet offers huge potential for computer vision tasks.

The EfficientNet model has been trained on Google's Colab. It is a Jupyter [45] based notebook service which provides free access to computing resources such as GPUs. Even though it is free to access it resources, yet there are some rules apply. Available resources can vary over time in Colab as fluctuations happen in demand. This means available GPU types, idle timeout period, maximum Virtual Machine lifetime vary time to time [46]. Available computing resources are usually Intel® Xeon® 2-core 2.2GHz CPU, 13GB RAM, 33GB HDD, Nvidia Tesla K80 GPU in free edition. Estimating the training time should be considered as Colab allows you to use Virtual Machine up to 12 hours and also will disconnect you from Virtual Machine if you are idle for too long. So it can be said smaller models are more suitable to be trained on Colab. But Colab still offers lots of functionality for free.

Colab offers following:

- Writing and executing code in Python
- Creating / Uploading / Sharing notebooks
- Importing/Saving notebooks from/to Google Drive
- Importing / Publishing notebooks from GitHub
- Importing external datasets
- Integrating TensorFlow, Keras, PyTorch, OpenCV

C. *Object Detection and Tracking*

During pre-processing stage, filters are implemented to reduce the noise level and enhance the quality of the depth stream [47]. Decimation filter has been applied to reduce the depth scene complexity effectively. Kernel size of the filter can vary from $2x2$ to $8x8$ pixels. While $4 - 8$ pixels are being selected for larger kernels, $2 - 3$ median depth value is selected for patches in regards to performance considerations. To preserve the aspect ratio, the image size is proportionally scaled down in both width and height. Given the example input size is $1280x720$ and scale factor is 3, the calculation would be $[1280, 720]/3$ -> $[426.6666667, 240]$ -> $[428, 240]$. To compensate changes in the resolution, the frame intrinsic parameters need to be recalculated after the new frame is produced. As decimation filter uses non-zero pixels, it also performs some hole filling operation.

The implementation of Spatial Edge-Preserving filter is based on Eduardo's paper [48]. The filter boosts the smoothness of the depth data by performing a series of 1D vertical and horizontal iterations. The key characteristics of the filter are, not being affected by parameters as it is linear-time compute and using high-order domain transform [47].

To improve the depth persistency, temporal filter manipulates per-pixel values based on the previous frames.

Temporal filter adjusts the depth values as well as updating the track history by performing a single pass on the depth data. In circumstances where the pixel data is invalid or missing, the filter decides whether the missing value should be corrected with stored data by using user defined persistency mode. The filter is best-suited for static scenes as it relies on historic data so smearing artefacts and visible blurring might be seen [47].

Holes Filling filter uses several methods to correct missing data in the stream. According to user defined rule, the filter receive four immediate pixel neighbours which are left, right, up, down pixels, and selects one of them [47]. The order of the applied filters [49] can be seen in figure 10.

Object detection is a computer vision technique in which a software system can detect, locate, and trace the object from a given image or video. The special attribute about object detection is that it identifies the class of object (person, table, chair, etc.) and their location-specific coordinates in the given image. The location is pointed out by drawing a bounding box around the object. The bounding box may or may not accurately locate the position of the object. The ability to locate the object inside an image defines the performance of the algorithm used for detection.

Detecting, locating and tracing the object in the given image or video are main attributes of the object detection software which is a computer vision technique. While detecting the desired object and its class in the given stream, it also identifies the object's location. By drawing a bounding box around the desired object, the location is being pointed out. In some cases, bounding boxes might not be accurate therefore post processing filters are used. The performance of the algorithm is defined by the ability of locating the desired object in the given stream [50].

To infer using custom trained PyTorch model, `.pth` file needs to be loaded using `torch.load(*.pth)` command. PyTorch [51] supports Nvidia GPUs for faster computation. Whether there is a CUDA supported GPU available can be learnt using `if torch.cuda.is_available():` command . If there is by typing `model = model.cuda()`, `*.pth` file can be sent to GPU. After having an input image, `scores, boxes = model(img.cuda())` command makes predictions and obtains the scores and bounding boxes of the desired object in the image.

To obtain stable results from the model, extended Kalman filter is applied after inference. Linear quadratic estimation also known as Kalman filter is a kind of algorithm which uses a series of measurements monitored over time and produces estimated values for unknown variables by approximating joint probability distribution for each time frame which is more accurate in contrast to a single measurement. One of the developers is Rudolf E. Kálmán and filter was named after him. Navigation, guidance and control of the vehicles are the common applications of Kalman filter [52]. Robotic motion planning, trajectory optimisation and robotic control are also main topics of Kalman filter [53]. The algorithm forms from two stages. The Kalman filter estimates the current state variables in the prediction stage along with their uncertainties. Once the next measurement is done which involves random noise, estimations can be updated with weighted average. As more weight is given to the algorithm, it can make the estimations with higher certainty.

Predicted state estimate:

$$x_{k|k-1} = f(x_{k-1|k-1}, u_k) \tag{1}$$

Predicted covariance estimate:

$$P_{k|k-1} = F_k P_{k-1|k-1} F_k^T + Q_k \tag{2}$$

As being a recursive algorithm, Kalman filter can run in real time. To make the optimal calculation, the Kalman filter assumes the noises are Gaussian. Regardless of being Gaussian, if the measurements and processes of covariances are known the Kalman filter is the best possible option for linear estimation with the minimum mean-square-error [54].

Measurement residual:

$$y_k = z_k - h(x_{k|k-1}) \tag{3}$$

Residual covariance:

$$S_k = H_k P_{k|k-1} H_k^T + R_k \tag{4}$$

Kalman gain:

$$K_k = P_{k|k-1} H_k^T S_k^{-1} \tag{5}$$

Generalisations and extensions also have been applied to the method such as the extended Kalman filter which works on nonlinear systems [55].

As being a recursive estimator, the Kalman filter needs current measurement and estimated state from previous time step to estimate the current state. No estimation history is needed contrast to batch estimation algorithms.

Updated state estimate:

$$x_{k|k} = x_{k|k-1} + K_k y_k \tag{6}$$

Updated covariance estimate:

$$P_{k|k} = (I - K_k H_k) P_{k|k-1} \tag{7}$$

Transition matrix:

$$F_k = \frac{\partial f}{\partial x}\Big|_{x_{k-1|k-1}, u_k} \tag{8}$$

Observation matrix:

$$H_k = \frac{\partial h}{\partial x}\Big|_{x_{k-1|k-1}} \tag{9}$$

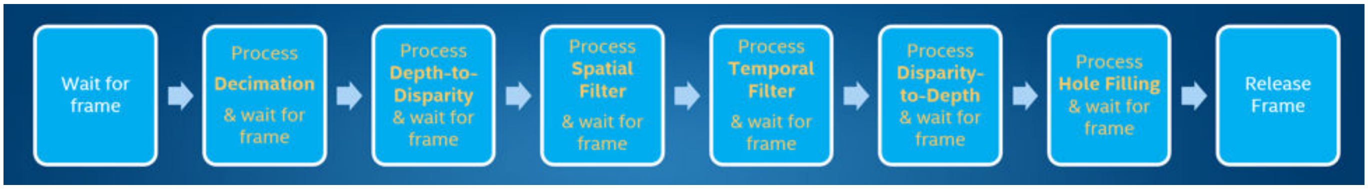
Wait for frame
Process Decimation & wait for frame
Process Depth-to-Disparity & wait for frame
Process Spatial Filter & wait for frame
Process Temporal Filter & wait for frame
Process Disparity-to-Depth & wait for frame
Process Hole Filling & wait for frame
Release Frame

Fig. 10. Filter flowchart

Two variables are used to represent the state of the filter. $x_{k|k-1}$ states the observations that has been estimated up to the time k given. $P_{k|k-1}$ states the covariance matrix which is a measure of the estimated accuracy.

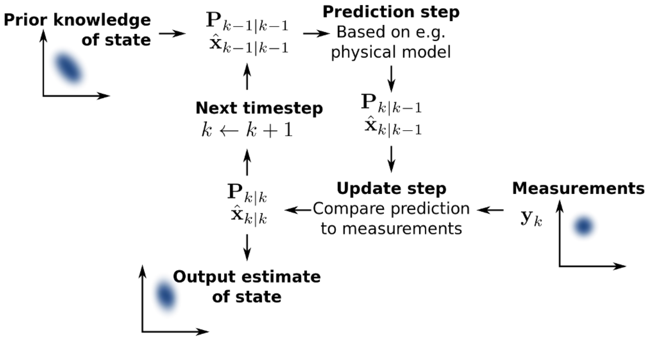

Fig. 11. The Kalman filter workflow

As shown in the figure 11 above, the Kalman filter has been adjusted to derive stable 3D world coordinates of the object. The formulas mentioned in the above belongs to 1D calculations. Extensions need to be done to the matrices. Therefore x_k is:

$$x_k = \begin{bmatrix} x \\ \dot{x} \\ \ddot{x} \\ y \\ \dot{y} \\ \ddot{y} \\ z \\ \dot{z} \\ \ddot{z} \end{bmatrix} \tag{10}$$

F and ***G*** matrices need to be augmented:

$$F = \begin{bmatrix} 1 & 0 & 0 & \Delta t & 0 & 0 & \frac{1}{2}(\Delta t)^2 & 0 & 0 \\ 0 & 1 & 0 & 0 & \Delta t & 0 & 0 & \frac{1}{2}(\Delta t)^2 & 0 \\ 0 & 0 & 1 & 0 & 0 & \Delta t & 0 & 0 & \frac{1}{2}(\Delta t)^2 \\ 0 & 0 & 0 & 1 & 0 & 0 & \Delta t & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & \Delta t & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & \Delta t \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{11}$$

$$G = \begin{bmatrix} \frac{\Delta t^2}{2} \\ \frac{\Delta t^2}{2} \\ \frac{\Delta t^2}{2} \\ \Delta t \\ \Delta t \\ \Delta t \\ 1 \\ 1 \\ 1 \end{bmatrix} \tag{12}$$

And ***R*** is:

$$R = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{13}$$

Where Δt is 0.001 which equals to 1000Hz. This represents the Kalman filter's frequency response.

D. 6D Pose Estimation

Detecting object's location and orientation forms the 6D pose estimation task which is important in robotic applications where the robot needs to be aware about the location of the object and estimate its pose to move towards to object for further operations [56].

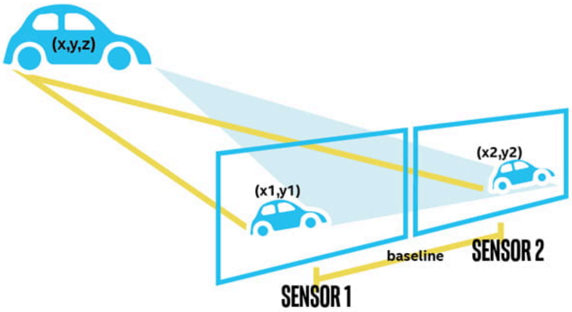

Fig. 12. Calculating the depth [57]

Stereo depth camera projects its infrared light onto a object to improve the accuracy of the data, unlike structured light camera, stereo camera can operate under any light condition to measure depth. Stereo depth cameras has two sensors with small space between. By taking two images from two sensor, as the distance is known between sensors, a stereo depth camera compares the images and uses the information to calculate the depth [57].

As stereo depth camera employs visual features of the object to measure, they work best in the most lighting conditions. The Intel® RealSense™ D435 cameras have additional an infrared projector which makes them to work in low lighting conditions without having problems. Using depth camera also interfering problem can be avoided in contrast to coded light camera which usually would have.

To be able to calculate object's pose, few parameters such as camera intrinsic, camera extrinsic and homography need to be obtained [58]. The camera calibration matrix usually called camera intrinsic is ***K***:

$$K = \begin{bmatrix} \alpha_u & \gamma & u_0 \\ 0 & \alpha_v & v_0 \\ 0 & 0 & 1 \end{bmatrix} \tag{14}$$

where α_u and α_v are the scale factor in (u, v) coordinates. γ is called skew where u and v are non-perpendicular.

External parameters or camera extrinsic is a matrix where ***R*** is a rotation matrix and ***t*** is a translation matrix.It represents the euclidean transformation from a world coordinate system to the camera coordinate system [58].

$$[R|t] = \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \end{bmatrix} \tag{15}$$

After successful detection in RGB stream, both streams can be aligned in order to obtain the object's real world coordinates from depth stream. Using `rs.align(rs.stream.color)` and `frameset.get_depth_frame()` commands, both colour and depth streams are being aligned. After alignment has been done, with `rs2_deproject_pixel_to_point` command real world coordinates can be derived from anywhere in the detection area as the centre of the object is usually being selected in most of the cases.

Calculated a set of n 3D points in real world coordinates and their corresponding 2D projections as well as the camera's intrinsic and extrinsic parameters, 6D pose estimation can be obtained. The perspective project model [59] is:

$$sp_c = K\left[R|T\right]p_w \tag{16}$$

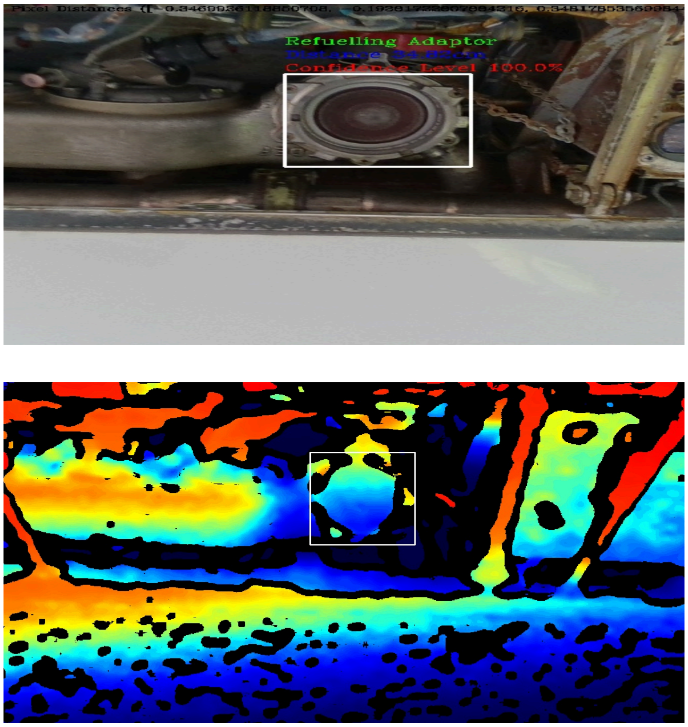
Pixel Distances ([-0.34699361188850708, -0.19381722007084216, 0.3481785356998
Refuelling Adaptor
Distance 34.82cm
Confidence Level 100.0%

Fig. 13. Aligned streams

where homogeneous world point is $p_w = [x\ y\ z\ 1]^T$, corresponding homogeneous image point is $p_c = [u\ v\ 1]^T$, f_x and f_y are the scaled focal lengths, γ is the skew. This leads to the equation for the model:

$$s\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & \gamma & u_0 \\ 0 & f_y & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \tag{17}$$

As necessary information is given to `cv2.solvePnP`, it produces the rotation and transformation vectors. A rotation vector is a compact and convenient representation of a rotation matrix. To be able to obtain Euler angles [60], rotation vector needs to be converted to rotation matrix using `cv2.Rodrigues`. The rotation matrix is shown as ***R***:

$$R = cos(\theta)I + (1 - cos\theta)rr^T + sin(\theta)\begin{bmatrix} 0 & -r_z & r_y \\ r_z & 0 & -r_x \\ -r_y & r_x & 0 \end{bmatrix} \tag{18}$$

From this point, converting rotation matrix to Euler angles is easy. And the formulation follows:

$$R_x(\psi) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & cos\psi & -sin\psi \\ 0 & sin\psi & cos\psi \end{bmatrix} \tag{19}$$

$$R_y(\theta) = \begin{bmatrix} cos\theta & 0 & sin\theta \\ 0 & 1 & 0 \\ -sin\theta & 0 & cos\theta \end{bmatrix} \tag{20}$$

$$R_z(\phi) = \begin{bmatrix} cos\phi & -sin\phi & 0 \\ sin\phi & cos\phi & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{21}$$

where ψ, θ and ϕ are the Euler angles [61].

V. Results

A. Object Detection Results and Analysis

The statistical criteria accuracy, precision, recall, specificity, F-measure and negative predictive value have been selected to quantitatively evaluate whether the succession of the PosEst method satisfies the autonomous ground refuelling approach for high standards.

Classification accuracy is the simplest metrics to show the model's success.By itself, it delivers the basic understanding of the results.

Accuracy:

$$A = \frac{TP + TN}{TP + TN + FP + FN} \tag{22}$$

Precision is a good indicator to show the model's learning rate on specific classes. In binary classification is indicates the model's response to non-class data.

Precision:

$$P = \frac{TP}{TP + FP} \tag{23}$$

Recall is another metric to differentiate the correct classification.

Recall:

$$R = \frac{TP}{TP + FN} \tag{24}$$

F scores are combining both precision and recall as being a better indication method. It is a harmonic mean of precision and recall.

F-measure:

$$F = \frac{(1 + \rho) \cdot P \cdot R}{\rho \cdot P + R} \tag{25}$$

Specificity is another popular metric that is being used.

Specificity:

$$S = \frac{TN}{TN + FP} \tag{26}$$

Negative predictive value is a proportion of negative results that have been classified truly. Where the model performs better, the negative predictive value is lower.

Negative Predictive Value:

$$NPV = \frac{TN}{TN + FN} \tag{27}$$

To evaluate the method, test set has been prepared using 81 refuelling adaptor images and 162 non-refuelling adaptor images. As can be seen in the figure 18, only 3 images have been misclassified. Only 1 image falsely predicted as a refuelling adaptor out of 243 images which is really low rate.

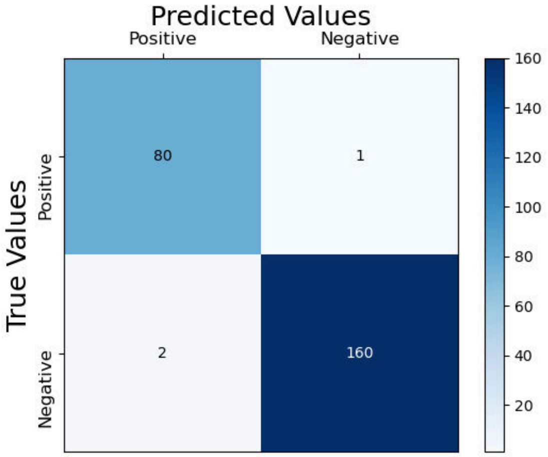

Fig. 14. Confusion Matrix

According to classification of test images, the results belong to precision, recall, F scores are calculated and shown below in the table.

TABLE I
METRICS TABLE

Accuracy %	98.7654321
Precision	0.975609756
Recall	0.987654321
F1-Score	0.981595092
F2-Score	0.97799511
Specificity	0.987654321
Negative Predictive Value	0.99378882

Overall results of the metrics are shown, custom trained model performs well on the test data. As the accuracy is not enough itself to evaluate the model, the other metrics such as precision, recall, F scores need to be considered. Especially having high precision and recall outcomes is showing the effective response of the model to any circumstances. Negative predictive value indicates the level of predicted false negatives are really low.

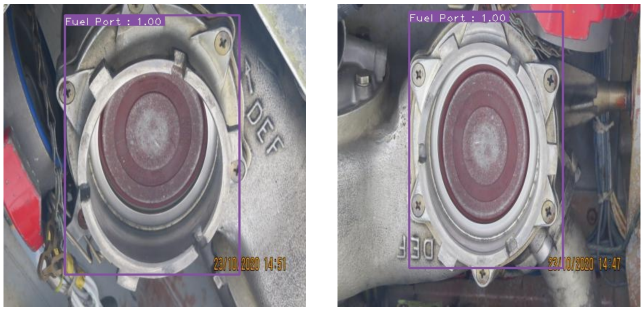

Fig. 15. Detections from close-up

It is relatively easy to detect an object from close-up. In the figure 15 above, refuelling adaptor can detected from close distance and different angles as well. It also shows the prediction rate is 100 which is quite high. In this case, the object is positioned 20 centimetres from the camera.

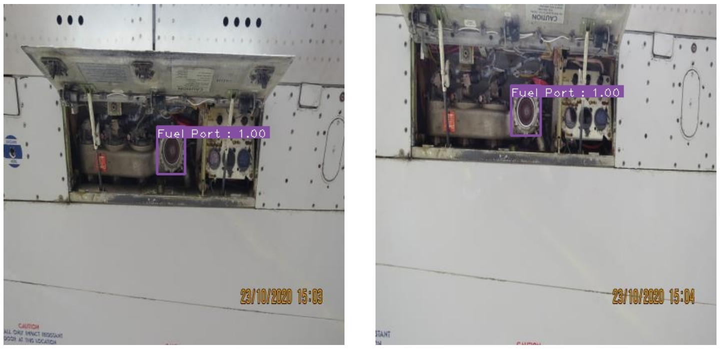

Fig. 16. Detections from distant

In most of the cases, machine learning models struggle to detect from far distances. In the figure 16, the refuelling adaptor which is located 2 metres apart from camera detected successfully. The model is also managed to obtain object's coordinates as well.

B. Pose Estimation Results and Analysis

Experiments have been carried out to analyse the results and determine the success of the PosEst method. Real world coordinates of the object x_0, y_0, z_0 are compared with calculated x, y, z using PosEst method. Absolute distance error and lateral error are the performance metrics for pose estimation stage. They are calculated as follow:

$$ADE = \left| \sqrt{x^2 + y^2 + z^2} - \sqrt{x_0^2 + y_0^2 + z_0^2} \right| \tag{28}$$

$$LE = \left| \sqrt{x^2 + y^2} - \sqrt{x_0^2 + y_0^2} \right| \tag{29}$$

To be able to measure the coordinates of the refuelling adaptor correctly, method needs to detect the refuelling adaptor in colour stream with high accuracy. Therefore, detected object's bounding box values need to be compared with its ground truth. The Intersection over Union provides a metric as the amount of predicted bounding box overlaps with the ground truth bounding box divided by the total area of both bounding boxes. Different Intersection over Union thresholds are defined starting at 0.5 and increasing to 0.95 by 0.05. In this case, 0.5 and 0.95 have been selected to evaluate the detections. This also an indication for correct coordinate derivation as the method is using both streams to obtain the coordinates. In this regards, accurate bounding box prediction is really important. The results belong the method can be seen in figure 17 and the metrics for bounding box $mAP@0.5$ and $mAP@0.95$ are measured as 0.996, 0.951 respectively. High accuracy in bounding box overlapping gives us accurate measurement in pose estimation. Constructing the refuelling adaptor in 3D space relies on the correct prediction in colour stream. Aligned streams in figure 13 has shown accurate construction of the object in 3D space as well.

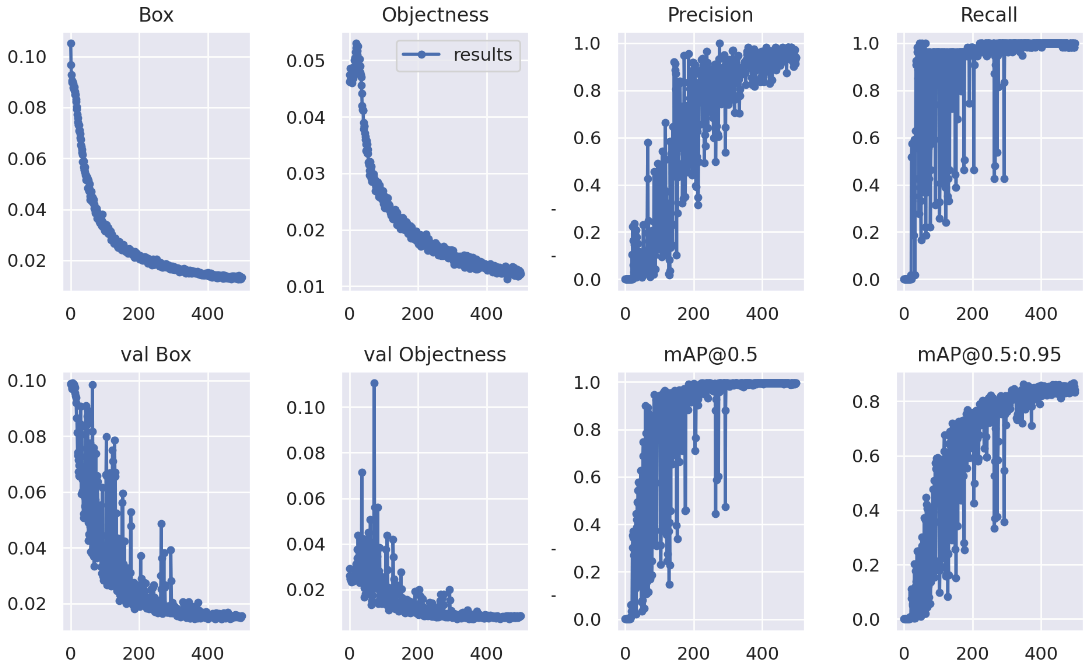

Fig. 17. The results of PosEst model

VI. Conclusions

This study presents a method ”PosEst” in order to facilitate 3D object grasping and 3D object tracking problems. Main moral of this study is to increase safety and efficiency by using autonomous systems in aircraft refuelling task. Aviation is one of the areas where high technology is being used. With the developing technology, it is predicted aviation will be subjected to a great digitalisation in the next 20 years. From the moment you enter the airport, smart security services, smart border controls and fully autonomous passenger aircrafts are the part of this change. With the development of technology, the need for people is decreasing in every field. The gap left by people can be filled by autonomous systems that could do their work tirelessly, more efficiently, faster and with higher precision. It has been observed the system can detect the refuelling adaptor and estimate its pose with high accuracy and precision. Currently the system is designed to detect the refuelling adaptor of Boeing 737-400 aircraft. Training set should be increased in order to work with other aircraft. After applying the Kalman filter instability in the system has been removed. However, there is a delay in the response of the system. This delay is not at a level that will affect the operation but this delay can be eliminated by working with alternative filtering methods.

VII. Acknowledgement

This project was supported by Republic of Turkey, Ministry of National Education through grant awarded to the first author. Dataset has been collected using Cranfield University’s Boeing 737-400 aircraft.

References

[1] Aircraft refueling procedures. (2018, April 16). Aircraft Engineer. https://www.aircraftengineer.info/aircraft-refueling-procedures

[2] Cummins, N. (2020, September 26). How Is An Aircraft Refueled? Simple Flying. https://simpleflying.com/how-is-an-aircraft-refuled/

[3] Refuelling with Passengers on Board. (2019, May 31). SKYbrary Aviation Safety. https://www.skybrary.aero/index.php/Refuelling_with_Passengers_on_Board

[4] Refuelling and Defuelling Risks. (2020, July 21). SKYbrary Aviation Safety. https://www.skybrary.aero/index.php/Refuelling_and_Defuelling_Risks

[5] Shipman, R. P. (1989). Visual servoing for autonomous aircraft refueling. AIR FORCE INST OF TECH WRIGHT-PATTERSON AFB OH SCHOOL OF ENGINEERING.

[6] Bottom Loading Adaptor. (2021, April 26). Liquip. https://www.liquip.com/products/aviation/aviation-bottom-loading-accessories/bottom-loading-adaptor/claval-bottom-loading-adaptor

[7] Junkins, J. L., Hughes, D., & Schaub, H. (2001). U.S. Patent No. 6,266,142. Washington, DC: U.S. Patent and Trademark Office.

[8] Valasek, J., Gunnam, K., Kimmett, J., Tandale, M. D., Junkins, J. L., & Hughes, D. (2005). Vision-based sensor and navigation system for autonomous air refueling. Journal of Guidance, Control, and Dynamics, 28(5), 979-989.

[9] Kimmett, J., Valasek, J., & Junkins, J. (2002, August). Autonomous aerial refueling utilizing a vision based navigation system. In AIAA Guidance, Navigation, and Control Conference and Exhibit (p. 4469).

[10] Kimmett, J., Valasek, J., & Junkins, J. L. (2002, September). Vision based controller for autonomous aerial refueling. In Proceedings of the International Conference on Control Applications (Vol. 2, pp. 1138-1143). IEEE.

[11] Tandale, M. D., Bowers, R., & Valasek, J. (2006). Trajectory tracking controller for vision-based probe and drogue autonomous aerial refueling. Journal of Guidance, Control, and Dynamics, 29(4), 846-857.

[12] Pollini, L., Mati, R., & Innocenti, M. (2004, August). Experimental evaluation of vision algorithms for formation flight and aerial refueling. In AIAA Modeling and Simulation Technologies Conference and Exhibit (p. 4918).

[13] Lu, C. P., Hager, G. D., & Mjolsness, E. (2000). Fast and globally convergent pose estimation from video images. IEEE transactions on pattern analysis and machine intelligence, 22(6), 610-622.

[14] Xufeng, W., Xinmin, D., & Xingwei, K. (2013, May). Feature recognition and tracking of aircraft tanker and refueling drogue for UAV aerial refueling. In 2013 25th Chinese Control and Decision Conference (CCDC) (pp. 2057-2062). IEEE.

[15] Wang, X., Kong, X., Zhi, J., Chen, Y., & Dong, X. (2015). Real-time drogue recognition and 3D locating for UAV autonomous aerial refueling based on monocular machine vision. Chinese Journal of Aeronautics, 28(6), 1667-1675.

[16] Chen, C. I., Koseluk, R., Buchanan, C., Duerner, A., Jeppesen, B., & Laux, H. (2015). Autonomous aerial refueling ground test demonstration—A sensor-in-the-loop, non-tracking method. Sensors, 15(5), 10948-10972.

[17] Chen, C. I., & Stettner, R. (2011, June). Drogue tracking using 3D flash lidar for autonomous aerial refueling. In Laser Radar Technology and Applications XVI (Vol. 8037, p. 80370Q). International Society for Optics and Photonics.

[18] Fischler, M. A., & Bolles, R. C. (1981). Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. Communications of the ACM, 24(6), 381-395.

[19] Yin, Y., Xu, D., Wang, X., & Bai, M. (2014). Detection and tracking strategies for autonomous aerial refuelling tasks based on monocular vision. International Journal of Advanced Robotic Systems, 11(7), 97.

[20] Wu, W., Wang, X., Xu, D., & Yin, Y. (2017). Position and orientation measurement for autonomous aerial refueling based on monocular vision. International Journal of Robotics & Automation, 32(1), 13-21.

[21] Gao, S., Cheng, Y., & Song, C. (2013). Drogue detection for vision-based autonomous aerial refueling via low rank and sparse decomposition with multiple features. Infrared Physics & Technology, 60, 266-274.

[22] Martínez, C., Richardson, T., Thomas, P., du Bois, J. L., & Campoy, P. (2013). A vision-based strategy for autonomous aerial refueling tasks. Robotics and Autonomous Systems, 61(8), 876-895.

[23] Martínez, C., Richardson, T., & Campoy, P. (2013, May). Towards autonomous air-to-air refuelling for UAVs using visual information. In 2013 IEEE International Conference on Robotics and Automation (pp. 5756-5762). IEEE.

[24] Martínez, C., Campoy, P., Mondragón, I. F., Sánchez-Lopez, J. L., & Olivares-Méndez, M. A. (2014). HMPMR strategy for real-time tracking in aerial images, using direct methods. Machine vision and applications, 25(5), 1283-1308.

[25] LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. nature, 521(7553), 436-444.

[26] Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. Advances in neural information processing systems, 25, 1097-1105.

[27] Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., ... & Fei-Fei, L. (2015). Imagenet large scale visual recognition challenge. International journal of computer vision, 115(3), 211-252.

[28] Yin, Y., Wang, X., Xu, D., Liu, F., Wang, Y., & Wu, W. (2016). Robust visual detection–learning–tracking framework for autonomous aerial refueling of UAVs. IEEE Transactions on Instrumentation and Measurement, 65(3), 510-521.

[29] Wang, X., Dong, X., Kong, X., Li, J., & Zhang, B. (2017). Drogue detection for autonomous aerial refueling based on convolutional neural networks. Chinese Journal of Aeronautics, 30(1), 380-390.

[30] Tan, M., & Le, Q. (2019, May). Efficientnet: Rethinking model scaling for convolutional neural networks. In International Conference on Machine Learning (pp. 6105-6114). PMLR.

[31] T. (2019, May 28). tensorflow/tpu. GitHub. https://github.com/tensorflow/tpu/tree/master/models/official/efficientnet

[32] EfficientNet: Improving Accuracy and Efficiency through AutoML and Model Scaling. (2019, May 29). Google AI Blog. https://ai.googleblog.com/2019/05/efficientnet-improving-accuracy-and.html

[33] The future of the airline industry — Airlines. (2017, August 29). IATA. https://airlines.iata.org/analysis/the-future-of-the-airline-industry

[34] Intel-RealSense-D400-Series-Datasheet-June-2020.pdf. (2018, January 1). Intel RealSense.

[35] Intel RealSense. (2021, May 5). Depth Camera D435 –. Intel® RealSenseTM Depth and Tracking Cameras. https://www.intelrealsense.com/depth-camera-d435/

[36] CLA-VAL 340AF : Pressure Fuel Servicing Adapter. (2019, September 24). Cla-Val. https://cla-val.ch/product/cla-val-340af-pressure-fuel-servicing-adapter/

[37] Department of Defense. (2014, September). Defense Standardization Program (DSP) Procedures (4120.24). https://www.esd.whs.mil/Portals/54/Documents/DD/issuances/dodm/412024m.pdf

[38] Wu, Z., Shen, C., & Van Den Hengel, A. (2019). Wider or deeper: Revisiting the resnet model for visual recognition. Pattern Recognition, 90, 119-133.

[39] Papers with Code. (2019, April 10). Papers with Code - EfficientNet Explained. https://paperswithcode.com/method/efficientnet

[40] Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., & Chen, L. C. (2018). Mobilenetv2: Inverted residuals and linear bottlenecks. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 4510-4520).

[41] Tan, M., Chen, B., Pang, R., Vasudevan, V., Sandler, M., Howard, A., & Le, Q. V. (2019). Mnasnet: Platform-aware neural architecture search for mobile. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 2820-2828).

[42] Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., & Fei-Fei, L. (2009). Imagenet: A large-scale hierarchical image database. In 2009 IEEE conference on computer vision and pattern recognition (pp. 248–255).

[43] Nilsback, M. E., & Zisserman, A. (2008, December). Automated flower classification over a large number of classes. In 2008 Sixth Indian Conference on Computer Vision, Graphics & Image Processing (pp. 722-729). IEEE.

[44] Krizhevsky, A., & Hinton, G. (2009). Learning multiple layers of features from tiny images.

[45] Bisong, E. (2019). Building machine learning and deep learning models on Google cloud platform (pp. 59-64). Berkeley: Apress.

[46] Carneiro, T., Da Nóbrega, R. V. M., Nepomuceno, T., Bian, G. B., De Albuquerque, V. H. C., & Reboucas Filho, P. P. (2018). Performance analysis of google colaboratory as a tool for accelerating deep learning applications. IEEE Access, 6, 61677-61685.

[47] Intel® RealSenseTM. (2019, April 23). Post-processing filters. Intel® RealSenseTM Developer Documentation. https://dev.intelrealsense.com/docs/post-processing-filters

[48] Gastal, E. S., & Oliveira, M. M. (2011). Domain transform for edge-aware image and video processing. In ACM SIGGRAPH 2011 papers (pp. 1-12).

[49] Intel® RealSenseTM. (2020, July 14). Depth Post-Processing for Intel® RealSenseTM Depth Camera D400 Series. Intel® RealSenseTM Developer Documentation. https://dev.intelrealsense.com/docs/depth-post-processing

[50] Zhao, Z. Q., Zheng, P., Xu, S. T., & Wu, X. (2019). Object detection with deep learning: A review. IEEE transactions on neural networks and learning systems, 30(11), 3212-3232.

[51] PyTorch. (2018, March 17). torch.cuda — PyTorch 1.8.1 documentation. https://pytorch.org/docs/stable/cuda.html

[52] Musoff, H., & Zarchan, P. (2009). Fundamentals of Kalman filtering: a practical approach. American Institute of Aeronautics and Astronautics.

[53] Ghysels, E., & Marcellino, M. (2018). Applied economic forecasting using time series methods. Oxford University Press.

[54] Humpherys, J., Redd, P., & West, J. (2012). A fresh look at the Kalman filter. SIAM review, 54(4), 801-823.

[55] Kalman, R. E. (1960). A new approach to linear filtering and prediction problems.

[56] Xiang, Y., Schmidt, T., Narayanan, V., & Fox, D. (2017). Posecnn: A convolutional neural network for 6d object pose estimation in cluttered scenes. arXiv preprint arXiv:1711.00199.

[57] Intel® RealSenseTM. (2020a, May 8). Beginner's guide to depth (Updated). Intel® RealSenseTM Depth and Tracking Cameras. https://www.intelrealsense.com/beginners-guide-to-depth/

[58] Signal Processing Stack Exchange. (2012, June 29). Step by Step Camera Pose Estimation for Visual Tracking and Planar Markers. https://dsp.stackexchange.com/questions/2736/step-by-step-camera-pose-estimation-for-visual-tracking-and-planar-markers

[59] Fischler, M. A., & Bolles, R. C. (1981). Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. Communications of the ACM, 24(6), 381-395.

[60] Euler, L. (1776). Novi commentarii academiae scientiarum petropolitanae. Nr, 20, 189-207.

[61] Slabaugh, G. G. (1999). Computing Euler angles from a rotation matrix. Retrieved on August, 6(2000), 39-63.