A Framework Based On Deep Learning Techniques For Multi-Drone ISR Missions Performance Evaluation In Different Synthetic Environments

Antenucci, L. Messina, A. Palumbo, S. Mazzaro, W. Matta
Vitrociset – a Leonardo Company SpA
00156, Rome
ITALY

a.antenucci@vitrociset.it; l.messina.somm@vitrociset.it; a.palumbo@vitrociset.it;
s.mazzaro@vitrociset.it; w.matta@vitrociset.it

ABSTRACT

This study aims to focus on how the synthetic world of today's simulators can act in synergy with that of neural networks and Deep Learning applied to video analysis, in particular using multi-drone/swarm systems for ISR missions. It is a fact that modern virtual engines used in both simulation and gaming have reached a level of realism that confuses a not so attentive observer. The question that arises spontaneously is, therefore, whether even an AI can be "cheated", and consequently the behaviour and decisions on-board the drone, changing the resulting actions of the fleet. That is, in more scientific terms, to evaluate whether the type and number of features that automatic learning systems on-board the drones, such as neural networks, can extract from synthetic images and can be reflected in the continuous world with significant advantages in the delicate and / or expensive phase of the training process, such as dataset creation and pre-exercise test. In fact, being able to model the elements in the simulated world at will, it is possible to reliably recreate situations and scenarios even impossible to recreate in real life (a network for the detection of lightning or explosions, for example), allowing the creation of datasets congruent in size according to a modern Deep Learning approach and reducing the physical times for the recovery of these images, also taking into account the constraints of the on-board computing power and capacity. Furthermore, it is worth asking whether the minor detail due to the discretization of the real scenario can act, in some circumstances, as a Principal Component Analysis (PCA) filter within the pre-processing of a dataset, during the dimensionality reduction process. The approach of the presented study will be experimental and will foresee a double direction of application. In a first phase we want to understand how neural networks trained on real datasets, on-board of one or more drones, behave in different synthetic environments. Three different simulators will be examined, namely VRForces, ROS Gazebo and VBS4, in order to also understand how the increasing of the graphical details will affect the accuracy and the precision-recall curve. The study presented in the proposed paper involves the area of AI object recognition and tracking, with particular focus on the problem of localization and therefore the detection of particular classes of objects such as people and vehicles. In a second phase of our research, the networks will be ready to be deployed taking into account the possible preparation of a hardware-in-the-loop use of COTS or custom autopilots, simulating real scenarios of ISR missions using a cooperative and intelligent fleet of drones. For this stage, our expertise is centred on a project called SWARM: a large industrial R&D Vitrociset project. It is an AI-Enabled Command and Control (C&C) system, able to execute and review ISR missions for mini/micro cooperative fleets of heterogeneous UAVs. SWARM will be used as testbed for the presented framework, testing and evaluating Deep Learning techniques for multi-drone ISR missions in different synthetic environments.

1.0 WHY USE A SYNTHETIC DATASET FOR SYNTHETIC ISR MISSIONS

A solution offered by the synthetic environment is that it allows to simulate infinite scenarios and weather /

light conditions that can be used to alleviate the low variability of many datasets, so as to study the behaviour of a network in extreme cases or not directly verifiable with real data. In particular, the execution of multi-drone missions involving the use of remotely piloted aircrafts, using a synthetic environment, allows to simulate events in conditions and operating scenarios that more and more faithfully reproduce those of real missions, and this proves to be particularly advantageous especially in the validation and testing phases, prior to any flight test phase. In this way, as well as being able to safely test all the implementation details, both at system and software level, it is possible to generate a consistent preliminary dataset on which to carry out the first analyses and an initial validation of the image processing algorithms. The main reasons for using this kind of approach are:

- The curse of dataset annotation, existing real datasets may lack precise annotations;
- Model evaluation, the use of synthetic datasets can bring out critical issues on the architecture, allowing the formulation of specific cases through the generation of a "controlled environment";
- Alleviation of bias, some datasets may not be able to better generalize all the possible cases to be subjected to a learning algorithm, synthetic datasets can therefore help cover any statistical flaws present within a real data collection;
- Solving problems related to privacy, the production of synthetic data can help overcome any obstacles related to the use of sensitive data.

Among the state of the art of modern frameworks we found Virtual Kitty. It is a framework born with the idea of mimicking the acquisition of videos and images within an urbanized scenario, exactly as it happened in its real counterpart, namely KITTI [1]: instead of making a real car travel with cameras, 3D scanners and laser, the same acquisition operations are performed in a virtual world within the Unity game-engine [2]. Among the aspects not to be underestimated, there is the fact that this type of acquisition makes it possible to accurately annotate the 3D and 2D bounding boxes, as the name suggests, they are spatial coordinates capable of allowing the identification of the area occupied by an object within a 3D space or a 2-dimensional area. On the other hand, other frameworks like SYNTHIA [3] place greater emphasis on the semantic annotation of objects in the virtual world: it offers 13 labelling classes, in order to have a large number of segmentation areas available. Here too, the urbanized area was generated through the use of Unity. One of the most interesting features of SYNTHIA lies in the fact that all the 3D objects have been made downloadable; consequently, it will be possible to generate new metropolitan areas in a completely random way, using each component as a single unit.

2.0 OUR VIRTUAL TEST BED AND EXPERIMENTATION FRAMEWORK

2.1 Virtual environment settings

For our study we made use of three different virtual environments able to receive data from the dynamic simulator of one or more COTS multirotor, which in turn (Figure 1) are linked to a standard PX4 autopilot whose behaviour is reproduced entirely through a Software-in-the-loop (SITL) approach. Through a special ground control station, the SWARM GCS [4,5], one of our products, it was possible to plan missions that use fleets of drones with different payloads. Being able to manipulate the scenario in which the UAVs can operate, it is easy to recreate the operating condition of interest in order to observe the world below from the camera in flight.

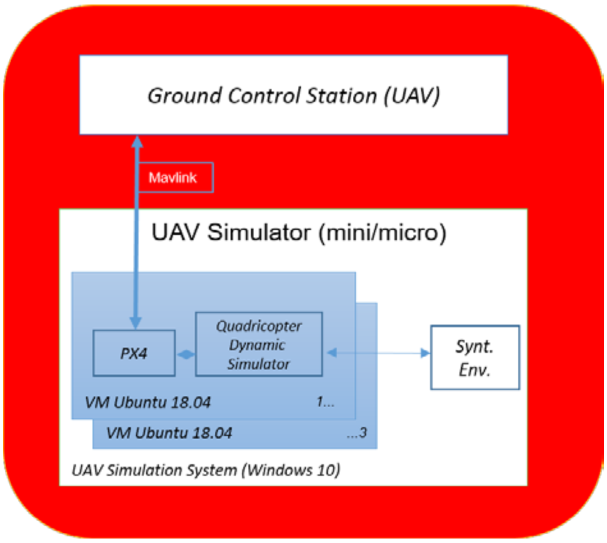

Ground Control Station (UAV)
Mavlink
UAV Simulator (mini/micro)
PX4
Quadricopter Dynamic Simulator
VM Ubuntu 18.04
1...
VM Ubuntu 18.04
...3
Synt. Env.
UAV Simulation System (Windows 10)

Figure 1: Simulation System Architecture

2.2 Using VBS as Synthetic Environment

One of the synthetic environments used to generate the scenarios is VBS4, developed by Bohemia Interactive Simulations (BISim), a simulator which, by interfacing with a database containing terrain elevation data and aerial images of the entire planet (Whole Earth Terrain), allows simulated missions to be carried out in any area of the Earth. Within the VBS4 simulation environment, we developed a plug-in that generates a synthetic scenario with one or more drones inside, each of which is equipped with a virtual camera capable of generating a video stream. The video streams are composed by frames obtained through a 3D rendering process of the scene observed from the point of view of the synthetic camera on board the drone.

The architecture used to simulate the behaviour of Multi mini/micro Drones for the purposes of the scenario described and analysed in this research activity, involves not only the use of the VBS4 simulation environment, but also of its special Software Development Kit (SDK). The VBS4 SDK provides an entire suite of APIs, so, through the use of these libraries, a C++ plugin was coded able to exploits various functionalities of the virtual environment. In particular, the following functionalities were exploited:

- The Scenario management;
- Entity management;
- Real Time Editor (RTE);
- Management and use of the Graphic Engine;
- Use of One World Terrain and buildings database;
- VBS Gateway (Indirectly used to enable HLA standard).

Once a scenario is started, the component generates 1 or more UAV entities at certain pre-set and configurable geographical coordinates (via a JSON file). These entities will represent the current state of the drone(s) managed in the Mini/Micro UAV Simulator and will thus allow the simulated drone to be dropped into a 3D scenario to perform mission simulations in contexts similar to the real-world mission context. In particular, the communication mechanism with the UAV Simulator is realized via a Publish/Subscribe message communication pattern.

The main functions that characterise the plugin are:

- Representation in the synthetic environment of the position, speed and attitude of the simulated UAV, the representation is made using a 3D model of the DJI Phantom 4 (Figure 2) drone (a 3D model of other drones in the VBS database could also be used or new ones generated);
- Generation of the scenes framed by the synthetic camera exploiting the Simulator's Graphic Engine functions;
- Sending of Synthetic Video Stream captured by the drone's video camera on the network;
- Management of several streams in the case of a multi-drone scenario;
- Manual control of the gimbal from the keyboard;
- Configurability of the number of drones and their position and of Video Stream characteristics;
- Scenario generation anywhere in the world;
- UAV damage management.

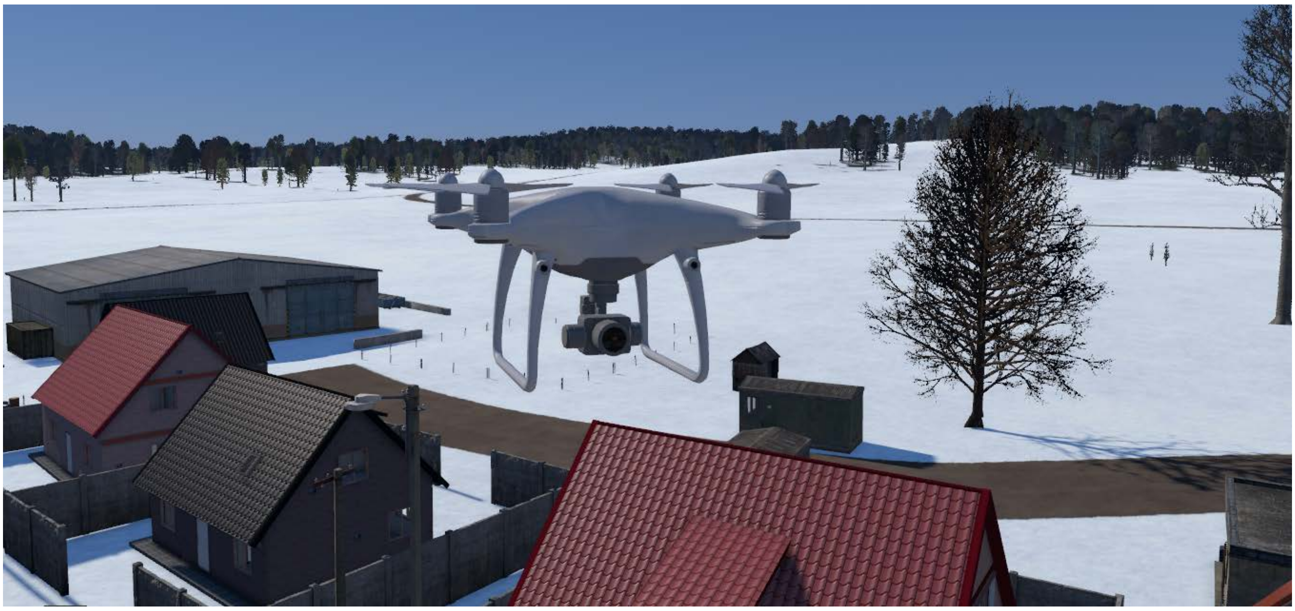


Figure 2: Model of the DJI Phantom 4 and detail of an urbanized scenario

In order to increase the level of detail of some scenarios set in urbanized contexts, VBS provides a database (divided by continents) of buildings. In this way, urbanized environments are recreated using datasets obtained from OpenStreetMap, and it is also possible to refine the 3D environment using VBS4's GEO modelling tool.

2.3 Using Gazebo as Synthetic Environment

Gazebo is a 3D simulator widely used in the field of robotics; is able to integrate simulations containing several robots in dynamic scenarios, simulating the behaviour of sensors and actuators and integrating physical parameters, dynamics of the contacts between objects and the effect of variable light, both in position and in intensity.

In Gazebo (Figure 3) it is possible to import, in addition to numerous objects, elements of the scenario and robots of various kinds already present in the default repository, three-dimensional models in the Collada format, which therefore allow a completely customizable simulation and the definition of simulation environments of various kinds, both indoor and outdoor. It also implements an editor that allows the construction of environments shown in real-time and usable in simulations. One of its main advantages is that it is open-source, making it easily accessible and enjoying the support of an extensive user base. This feature has led Gazebo to be integrated into the SITL (Software-in-the-Loop) simulation module of the PX4 autopilot, thus making it immediately available for testing software and algorithms designed for the real world. However, while the simulation of the sensors of the IMU (Inertial Measurement Unit) can be easily obtained with mathematical calculations "tricking" the autopilot, designed to interact with real drones, the simulated images can be much less realistic and not suitable for use with AI algorithms, leading to unpredictable results in the transition from synthetic environment to real world. This problem was marginal in the study of mainly static situations but can be decisive by applying more sophisticated algorithms, for example in the identification of people and crowds with aerial images. The images were acquired using an Iris drone (available in the PX4 repository) equipped with IMU and a camera configurable via file and implemented as a C ++ plugin.

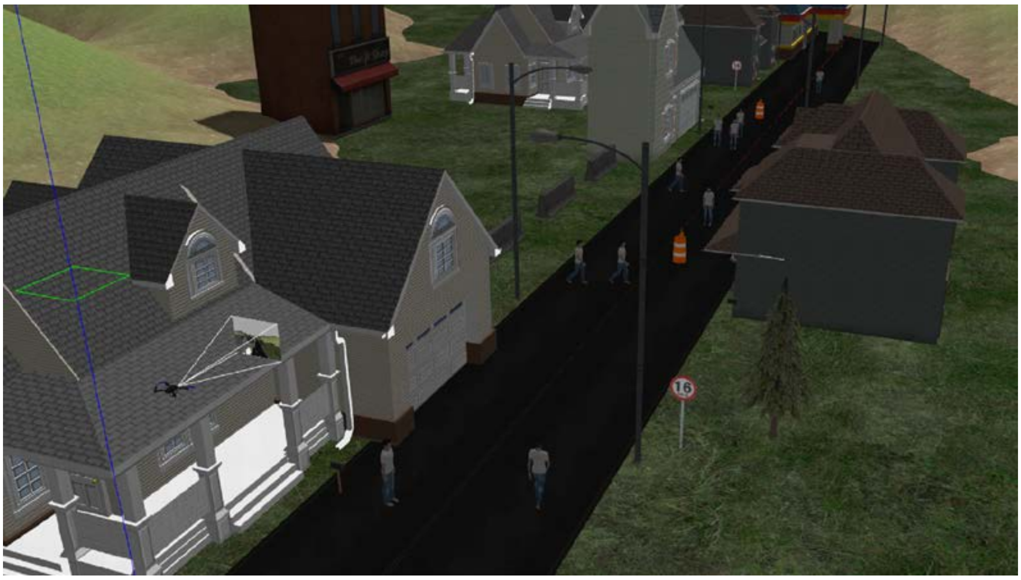


Figure 3: An example of a Gazebo ISR Urbanized Scenario

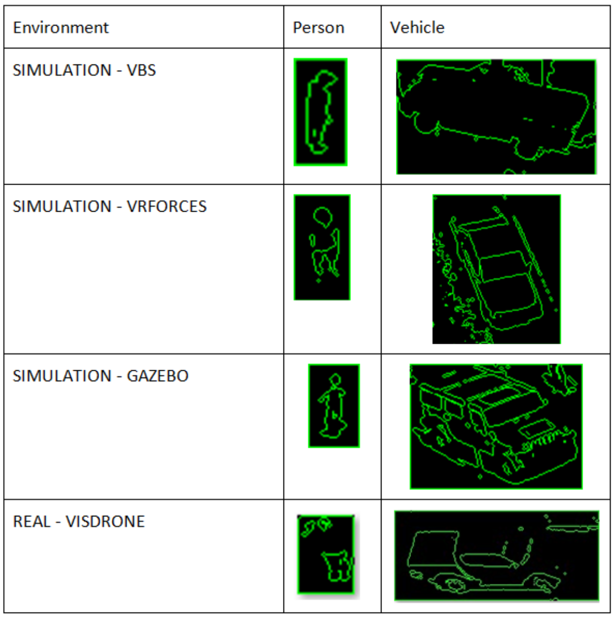
To allow the use of drones integrated into the Vitrociset SWARM [4,5] system on multiple simulation platforms, a software has been developed that simulates their dynamics. It interacts with the SITL module of the PX4 autopilot by generating fictitious measurements of the IMU and GPS sensors. It allows the simulation of mini / micro category multirotor drones equipped with Pixhawk flight controller and PX4 firmware, configurable in their main physical characteristics, such as inertia matrix, weight, diameter and rotor arrangement.

2.4 Using VR-Forces as Synthetic Environment

For the tests through VR-Forces, a plugin was created using the SDK of the framework that allowed us to move a particular type of rotary wing UAV, whose dynamics have been adapted and the communication in terms of commands and flow acquisition has been rewritten. In particular, a specific use case was defined, i.e., the flight of multi drones in an urbanized context characterized by the presence of houses, roads, vehicles, people and vegetation. The multi-drone flight plan has been made compliant with the specifications of the particular UAV model used and the simulation environment itself, guaranteeing a comparable point of view in the three virtual environments. The parameters of the cameras have been appropriately calibrated in such a way as to guarantee the same display from the sensor. In particular, the pinhole model was used to describe the image of the video cameras, obtaining a function that defines a mapping between the points of the three-dimensional space and the detected pixels. When not known or configurable, the intrinsic parameters and the radial distortion values were obtained following the algorithm of Zhang [6]. By exploiting a set of images containing a known geometry (a chessboard), this algorithm is able to extract an estimate of the parameters which improves as the images supplied in input grow.

3.0 COMPARISON BETWEEN THE SYNTHETIC AND THE REAL ENVIRONMENT

It is important to underline the fact that the use of synthetic images alone risks introducing new biases, due to some repeated patterns that could compromise the variability of the new dataset. Taking visual data as an example, it was noted how important a realistic rendering is to mitigate the shock resulting from domain translation: this problem is known as domain adaptation [7]. The use of a synthetic, and therefore discreet, environment in the generation of images has an evident effect in the ease with which the outlines of objects can be identified. In order to better understand the characteristics between synthetic and real datasets, preliminary studies were carried out on the contents of the bounding boxes relating to people and vehicles. In particular, classical computer vision and image manipulation techniques were investigated in identifying the main differences between the objects detected in the images of the VISDRONE dataset (real) and those identified in the various simulators (VBS, VRForces and Gazebo).



Environment	Person	Vehicle
SIMULATION - VBS		
SIMULATION - VRFORCES		
SIMULATION - GAZEBO		
REAL - VISDRONE		

Figure 4: Person and Vehicle classes on which a mask has been applied to obtain the outlines

Proceeding with the extraction of the contours of the two classes of objects, it soon became clear that in the continuous the image is noisier and subject to blur and the edges, consequently, are less defined and, therefore, more difficult to detect (Figure4). The greater simplicity in object recognition, however, is achieved net of a significant loss of "real" information and does not allow the total replacement of the datasets with synthetic data.

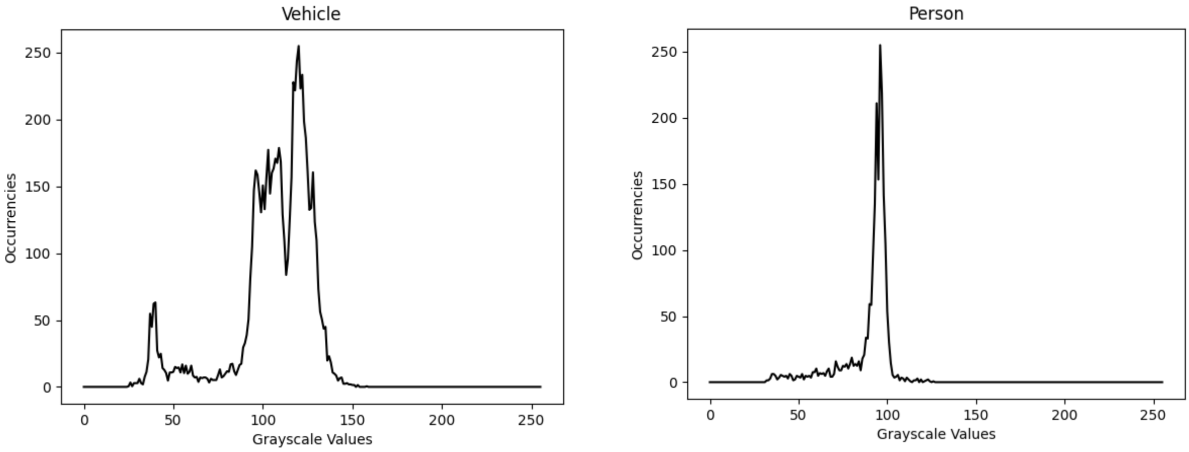


Figure 5: Person and Vehicle histogram representing grayscale values occurrences in VRForces

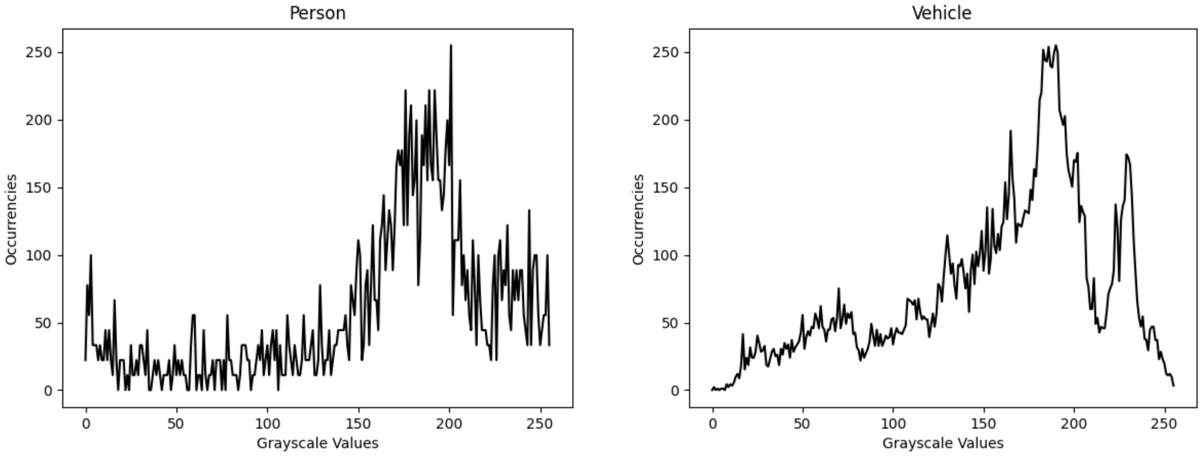


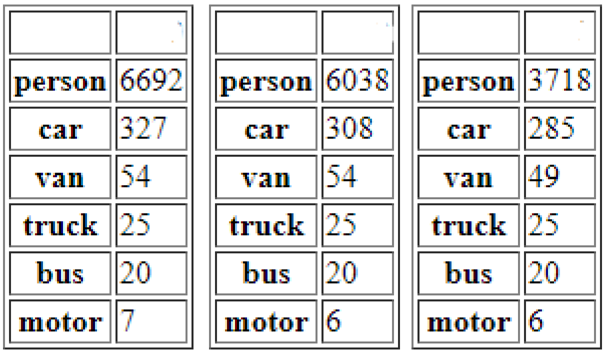
Figure 6: Person and Vehicle histogram representing grayscale values occurrences in visdrone real data

Figure 5 and Figure 6 show how, by applying appropriate masks to remove backgrounds and analysing the information content of the pixels of people and vehicles, the histogram of recognized objects, converted to grey scale, shows a more uniform distribution of information content in the real case, while in the synthetic case (of which VRForces is depicted but the same applies to the other two) we can find a more marked and restricted distribution, due to the "discretization" of the synthetic domain. Therefore, net of a gain in the simplicity of the image, a loss of information was obtained compared to the real case. It should be noted that the concentration of the pixel content towards a lower value in the synthetic case is also due to a brightness condition different from the real case.

3.1 Dataset and Video Streaming Engineering

Once the synthetic video streams were acquired, the individual frames were extracted and the annotation files related to them were generated. To do this, an automatic procedure has been developed capable of "selftagging" the images through an active learning technique, through the configuration of the different neural networks, and the mapping of the ids with the label. Obviously, having a neural network capable of fulfilling the task of identifying and labelling the bounding boxes in the images, we have one part of the problem solved by default, which is why a pool of neural networks was considered among the models provided by the tensor flow framework. Once the prediction has been made for each model, the calculated scores were aggregated according to a suitably weighted cost function such as to provide a unique result that will be considered as an oracle. The algorithm exploits both the type of label of the bounding boxes and their position and takes into account the Intersection Over Union (IOU) between them to discard, move or confirm the final annotation. The method used is based on an implementation taken from the paper by C. Brust [8]. Only in a second phase did we manually intervene on the bounding boxes of ground truth, using the label Image software.

Once the ground truth was set, three versions of TFRecords (standard data format for Tensor Flow) have been generated, where in test-set A no filter was applied, in B a 100 Pixel BB area filter and in C a 200 Pixel BB area filter. Table 1 shows the detail of the number of objects per class in the three test sets.



person	6692
car	327
van	54
truck	25
bus	20
motor	7

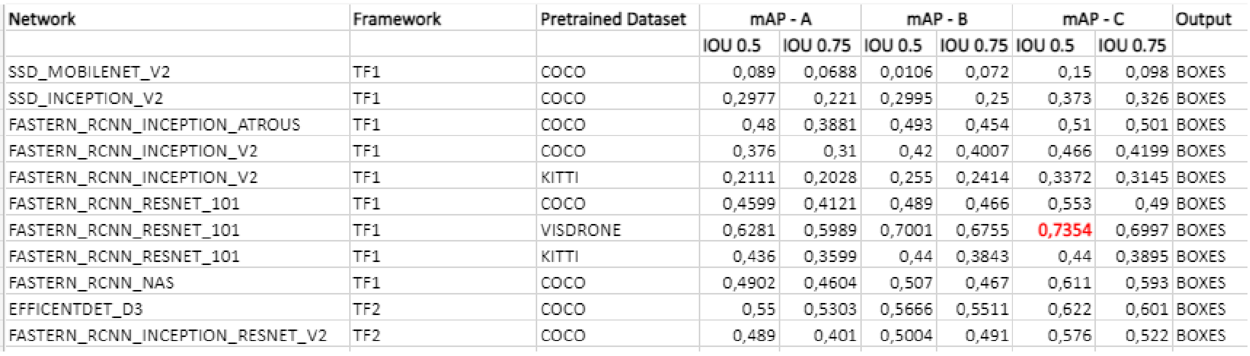
person	6038
car	308
van	54
truck	25
bus	20
motor	6

person	3718
car	285
van	49
truck	25
bus	20
motor	6

Table 1: test-set A (left), B (center) and C (right) composition

3.2 Our Results and Conclusions

Leveraging the most popular metrics for behavioural assessment, and in particular the accuracy, of object detectors (AP, mAP, precision, recall), it is possible to analyse the table below with the results obtained on the three test-sets previously described with the only modifications of having merged the classes of the vehicle subtypes into a single category (thus obtaining only two classes, persons and vehicles) and the reset of the images (possibly adding a padding in order not to introduce distortion) according to the size of the input batch to the different deep neural networks used. Different frameworks and different pre-treated architectures were investigated on public datasets corresponding to the use case of interest. The values are reported as the IOU varies but only for the threshold value that led to the best results and which corresponds to a value of 0.4 as can be seen in Table 2:



Network	Framework	Pretrained Dataset	mAP - A		mAP - B		mAP - C		Output
			IOU 0.5	IOU 0.75	IOU 0.5	IOU 0.75	IOU 0.5	IOU 0.75	
SSD_MOBILENET_V2	TF1	COCO	0,089	0,0688	0,0106	0,072	0,15	0,098	BOXES
SSD_INCEPTION_V2	TF1	COCO	0,2977	0,221	0,2995	0,25	0,373	0,326	BOXES
FASTERN_RCNN_INCEPTION_ATROUS	TF1	COCO	0,48	0,3881	0,493	0,454	0,51	0,501	BOXES
FASTERN_RCNN_INCEPTION_V2	TF1	COCO	0,376	0,31	0,42	0,4007	0,466	0,4199	BOXES
FASTERN_RCNN_INCEPTION_V2	TF1	KITTI	0,2111	0,2028	0,255	0,2414	0,3372	0,3145	BOXES
FASTERN_RCNN_RESNET_101	TF1	COCO	0,4599	0,4121	0,489	0,466	0,553	0,49	BOXES
FASTERN_RCNN_RESNET_101	TF1	VISDRONE	0,6281	0,5989	0,7001	0,6755	0,7354	0,6997	BOXES
FASTERN_RCNN_RESNET_101	TF1	KITTI	0,436	0,3599	0,44	0,3843	0,44	0,3895	BOXES
FASTERN_RCNN_NAS	TF1	COCO	0,4902	0,4604	0,507	0,467	0,611	0,593	BOXES
EFFICENTDET_D3	TF2	COCO	0,55	0,5303	0,5666	0,5511	0,622	0,601	BOXES
FASTERN_RCNN_INCEPTION_RESNET_V2	TF2	COCO	0,489	0,401	0,5004	0,491	0,576	0,522	BOXES

Table 2: Our Results with different Deep Neural Networks

As it was logical to expect, the networks trained on test-set C without a bounding box with a lower area of 200 pixels performed on average better than the other two. The best obtained corresponds to the fastern_rcnn_resnet_101 pre-trained on the visdrone dataset, a result which is also predictable given that the use case considered, that is the scenarios created on the different simulation environments (which portray scenes taken from a camera on board the drone in flight), it is more similar to the original one. Furthermore, on this configuration it is possible to note how the delta of mAP parameter between the different datasets is lower than in the other cases, this consideration is reflected in the fact that the visdrone dataset has a high wealth of information content even for small bounding boxes, as shown in Figure 7:

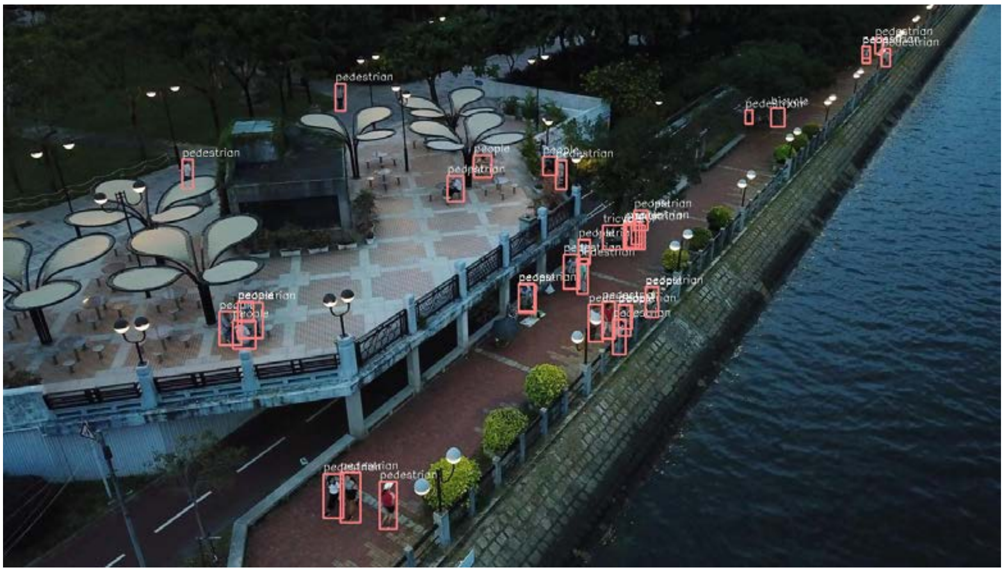
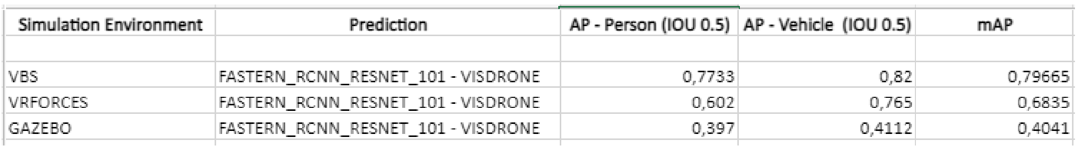


Figure 7: Example of annotations in visdrone dataset in a Multi-drone ISR Mission

A further reflection can be made regarding the quality of the graphic rendering of the three scenarios, analysed in the paragraph 3, and the fact that the best APs on the individual classes were obtained precisely in the portion of the dataset (without filters) corresponding to the VBS use case simulator.



Simulation Environment	Prediction	AP - Person (IOU 0.5)	AP - Vehicle (IOU 0.5)	mAP
VBS	FASTERN_RCNN_RESNET_101 - VISDRONE	0,7733	0,82	0,79665
VRFORCES	FASTERN_RCNN_RESNET_101 - VISDRONE	0,602	0,765	0,6835
GAZEBO	FASTERN_RCNN_RESNET_101 - VISDRONE	0,397	0,4112	0,4041

Table 2: Synthetic Environments Comparison

It is believed that this difference is also accentuated by the fact that VBS was also the simulation environment that provided a greater number of "person" category annotations and that therefore, even if only for purely statistical reasons, the metrics on the average of misses' detections or false positives is less prone to the problems of "small numbers". That said, it is evident that the use of a synthetic environment equipped with the most performing graphics engine of the three guarantees better results even by reporting the number of pixels of the videos obtained from it at the same value as the others.

3.3 Future Work

The work and the results obtained in the first part of our study allowed us to identify the best neural network for the application scenario of interest (i.e., object recognition from a camera on board one or more drones) and also a good starting point for the creation of a synthetic dataset. In the second phase of the experimentation, it will be possible to investigate the inverse field of application compared to the previous one, namely the use of synthetic (and mixed synthetic / real) datasets for the training of neural networks and their continuous use.

REFERENCES

[1] Gaidon, A. – Wang, Q. – Cabonn, Y. – Vig, E. "VirtualWorlds as Proxy for Multi-object Tracking Analysis." 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2016)

[2] Geiger, A. – Lenz, P. – Stiller, C. – Urtasun, R. "Vision meets robotics: the KITTI dataset." The International Journal of Robotics Research. 32. (2013)

[3] Ros, G, – Sellart, L. – Materzynska, J. – Vazquez, D. – Lopez A.. "The SYNTHIA dataset: A large collection of synthetic images for semantic segmentation of urban scenes." Proceedings of the IEEE conference on computer vision and pattern recognition. (2016)

[4] L. Messina, S. Mazzaro, A. E. Fiorilla, A. Massa, W. Matta, "Industrial implementation and performance evaluation of LSD-SLAM and map filtering algorithms for obstacles avoidance in a cooperative fleet of unmanned aerial vehicles", 3rd International Conference on Intelligent Robotics and Control Engineering, IRCE2020.

[5] A. Antenucci, S. Mazzaro, A. E. Fiorilla, L. Messina, A. Massa, W. Matta, "A ROS Based Automatic Control Implementation for Precision Landing on Slow Moving Platforms using a Cooperative Fleet of Rotary-Wing UAVs", 2020 5th International Conference on Robotics and Automation Engineering, ICRAE2020.

[6] Zhengyou Zhang. A Flexible New Technique for Camera Calibration, Microsoft Research, One Microsoft Way, 1998.

[7] Nikolenko, S. "Synthetic Data for Deep Learning." ArXiv abs/1909.11512 (2019).

[8] C. Brust, C. Käding and J. Denzler (2019) Active learning for deep object detection, VISAPP