

Human Detection for Search and Rescue Applications with UAVs and Mixed Reality Interfaces

Raúl Llasag^{1,2}, Diego Marcillo¹, Carlos Grilo^{2,3}, Catarina Silva^{2,4}

¹Departamento de Ciencias de la Computación, Universidad, ESPE, Sangolquí, Ecuador

²Escola Superior de Tecnologia e Gestão, Instituto Politécnico de Leiria, Leiria, Portugal

³CIIC – Centro de Investigação em Informática e Comunicações, Instituto Politécnico de Leiria, Leiria, Portugal

⁴CISUC - Centro de Informática e Sistemas da Universidade de Coimbra, Portugal

{rhlilasag, dmmarcillo}@espe.edu.ec; {carlos.grilo, catarina}@ipleiria.pt

Abstract — Human Detection (HD) in natural disaster areas with the use of Unmanned Aerial Vehicles (UAV) or drones is an important approach to preserve human lives in Search and Rescue (SAR) operations, allowing to rescue injured people through previous and fast exploration from safer zones. Nevertheless, constraints on artificial vision and interactive interfaces remain a challenge, namely the ones related to Computer Vision (CV) techniques, since HD cameras drones need to transmit video streams in real-time to Ground Stations (GS). Additionally, GS interfaces of this type of applications are usually not easy to use for unexperienced pilots. In this work, we propose an architecture for SAR operations, integrating an interactive interface that can be used with commercial UAV and Head-Mounted Display (HMD) devices, which provide Mixed Reality (MR) for GS. Results on an application prototype show that a fine tuning process of a non-traditional object detection model with SAR images can improve human detection precision with real-time video streaming.

Keywords – Human Detection; Unmanned Aerial Vehicles; Search and Rescue; Computer Vision; Mixed Reality; Head-Mounted Display.

I. INTRODUCTION

Through the years, efforts have been made so that faster human assistance on natural disasters can be provided. In this regard, the use of Unmanned Aerial Vehicles (UAV) is being explored as a way of quickly providing information about the actual situation of affected zones. Additionally, for automatized people location, Computer Vision (CV) techniques can be used to search humans on video frames captured with the camera's UAV [1].

In general, the video is transmitted from the UAV to the Ground Station (GS), which then uses detection algorithms. However, Human Detection (HD) with UAVs is not an easy task because it has associated problems, such as poor detection when the cameras rotate over their axis, when people take different postures, when the human body is incomplete on the image, poor scene illumination, among others [2].

In [3], [4] and [5], poor detection issues were partially solved using traditional machine learning techniques and training models with single class datasets (see Section VI) combinations. [2] provide better detection results using the same single class datasets combinations and deep learning

techniques. However, these authors mention that there is a tradeoff between accuracy and processing speed.

Besides that, [6] mentions that beginner pilots have trouble interacting with GS interfaces that usually provide 2D screens with complex tasks. In an attempt to mitigate this problem, Head-Mounted Displays (HMD) for GS interfaces with 3D augmented reality are used in [6]. In [2], a smart-glass is proposed to train beginner pilots, but the proposed approach was only tested on simulated environments. In [6] and [7] Virtual Reality (VR) interfaces display images captured from the UAV and its route over a map. However, with the last interface, pilots can not directly see the UAV, resting them to be aware of the UAV real position during the flight.

To our knowledge, until today, no project has proposed an architecture for SAR with HD that allows pilots to be aware of UAVs real-position and controlling them with a Mixed Reality (Augmented Reality + Virtual Reality) application installed on a MR headset or on a HMD. In this paper, we propose an architecture that addresses the problems mentioned above.

The rest of this paper is structured as follows: In Section II, SAR operations are described. In Section III, commercial drones and MR headsets models are introduced. Then, in Section IV, the main features of the system architecture are detailed. Next, in Section V, object detection approaches and their object detection metrics are presented. In Section VI, human detection over streaming video is detailed according to the constraints found. In Section VII, a fine-tuning process explains how detection accuracy can be improved. Finally, conclusions on the challenges tackled for a successful application and future works are present in Section VIII.

II. SEARCH AND RESCUE OPERATIONS

In [8], a rescue scenario is described in two parts of a larger problem. The first one is related to the identification and location of injured people and the second one consists in delivering supplies or resources.

However, it is mentioned that only the first problem is tackled due to the physic characteristics of commercial drones. Table 1 shows the advantages and disadvantages of different types of drones.

TABLE I. DRONE TYPES

Drone type	Advantages	Disadvantages	
Fixed-wing	Large area coverage	Inconvenient launch and landing	Price
Helicopter	Hover flight	Harder to flight than multicopter	
Multicopter	Availability price Hover flight	Low payload Short flight duration	

According to [9], the use of drones for SAR operations is recommended only on climate conditions with low communication interferences. It is also, mentioned that fixed-wing drones quickly provide knowledge about affected zones while rotary-wing drones can stabilize on the air for a change of flight direction, making more detailed inspections, for example, human detection.

Other considerations are the type of SAR application and the drone's flying complexity. In [8], [10] and [11] non-commercial helicopters were piloted by experts and had successful SAR missions, while in [6] and [12] commercial quadcopters were easily piloted, because these provide characteristics such as mission planning, take-off, landing and gimbal movements (roll, pitch, yaw). Our project is focused on Urban SAR operations with the use of commercial drones because they provide software development kits (SDK), assistance to beginner users and intelligent flight modes.

III. COMMERCIAL DRONES AND MR HEADSETS

We propose a general architecture to be implemented using tools of semi-autonomous drones and a MR headset because semi-autonomous drones minimize the effort needed by inexperienced pilots in order to accomplish complex tasks. Also, there are software development tools for MR headsets that allow the creation of interactive interfaces.

A. Semi-autonomous drones

In [13], DJI¹ and Parrot² are mentioned as the two first companies on sales position. Parrot offers only VR experiences over HMD with the BEBOP-2³ model, while DJI with 70% of the global market does not have a HMD included in their model's boxes. The success factors of these drone models are semi-autonomous control, low prices and integration with the developer community thanks to software development kits for Android and iOS mobile applications.

B. Mixed Reality headsets

Mixed Reality interfaces for headsets started with the HoloLens⁴ model created by Microsoft in 2015. The purpose of these devices was mixing Virtual Reality (VR) with Augmented Reality (AR) through 2D and 3D hologram projections. Today, other VR/AR models are present on the market such as the second version of HoloLens, Magic Leap⁵ and Meta 2⁶. The HoloLens headset differs from other head-mounted displays (HMDs) on the fact that it provides spatial

computing and user mobility. In other words, HoloLens runs its operating system on the display hardware and projects holographs based on a spatial mapping. On the other hand, Magic Leap handles the primary data and graphics processing with a spatial mapping approach on a small hip-mounted computer, while Meta 2 needs a computer connected for data processing. However, all these headset models allow creating MR interface applications with hand gestures and voice using the Unity engine.

IV. SYSTEM ARCHITECTURE IMPLEMENTATION

Figure 1 illustrates the SAR architecture system proposed and implemented in this work. It consists of three applications. The first one uses methods to configure and control the UAV (UAV-App), the second one is a Mixed Reality application (MR-App) and the third one is a Communication Central Server (CCS) that acts as an interpreter between the UAV-App and the MR-App.

Architecture features and mechanisms implemented are presented in three main parts. The first details the real-time data sharing and data persistence necessary to configure and control the drone flight. The second details the rendering, presentation modules on the MR-App and location awareness. Finally, the third part details video transformations carried out to provide streaming video with computer vision

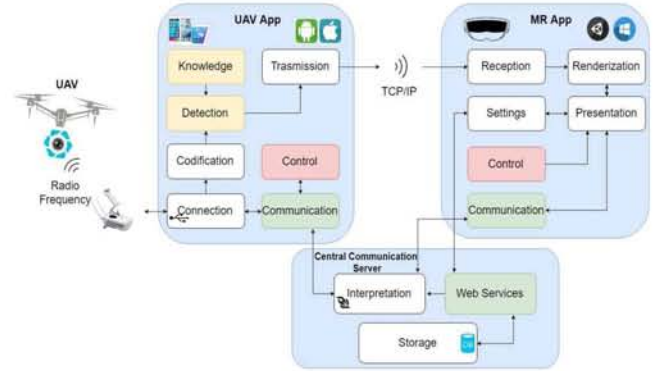


Figure 1. SAR architecture

A. Real-Time Data Sharing and Data Persistence

Real-time data sharing definitions are implemented on the communication and settings modules in the UAV-App and the MR-App. These conditions were defined in order to implement an energy and battery saving mechanism and are illustrated in Figure 2.

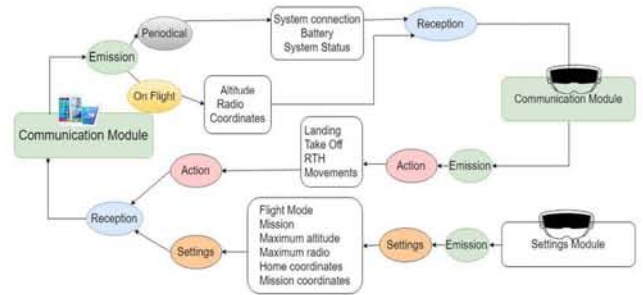


Figure 2. Real-Time Data Sharing

¹ <https://www.dji.com>

² <https://www.parrot.com/global/drones>

³ <https://www.parrot.com/global/drones/parrot-bebop-2-fpv>

⁴ <https://www.microsoft.com/en-us/hololens>

⁵ <https://www.magicleap.com>

⁶ <https://www.metavision.com>

In this project, a battery saving mechanism specifies the emission and reception of data only when sensor data is updated or when an event is activated on the control and settings modules. The data emission from a UAV-App could be done only when the system or sensor status need to be updated after receiving data on the MR-App interface. However, data emission is also activated when the drone is flying, for drone's location transmission.

On the other hand, the MR-App emits data according to flight action events or setting events. In the first case, data comes from the *Communication* module, while in the second case, data comes from the *Settings* module. This module interchanges data with the *Storage* module located on the CCS throughout a *Web service* module in order to save system configurations.

Information corresponding to the configuration, such as, flight distance allowed, origin position, GPS mission coordinates, pilot experience and mission type need to be saved in storage through the *Web service* module and the *Storage* module to save drones configurations and mission planning routes. We tested with a Phantom 4 drone model and the first version of a MR headset called HoloLens, as illustrated in Figure 3.



Figure 3. Phantom 4 controlled by HoloLens

B. Mixed Reality Interface

The holograms projected in Figure 4 are texture instances located on the *Presentation* module that loads data from the *Control*, *Settings*, *Communication* and *Rendering* modules. This data is directly updated when the *Control* and *Settings* modules events are activated. However, this data may also be indirectly updated when it is received from the *Communication* module through an observer pattern implementation or when it is received from the *Rendering* module in bitmap format, providing video streaming service and location awareness when the drone flies long distances and the pilot cannot directly see it.



Figure 4. Mixed Reality or VR/AR interface

C. Streaming video with Computer Vision

Images projection on the reality, such as illustrated in Figure 4, renders bitmaps transmitted from the *Transmission* module to the *Reception* module throughout a TCP/IP socket connection. However, this connection type could experiment disconnections on a poor signal of a wireless network. For this reason, *Transmission* and *Reception* modules implement a channel reconnection mechanism. This mechanism is supported by connection and disconnection events implemented on the *Interpretation* module.

Besides disconnections, video transmissions could also suffer interruptions when buffers have used all the memory buffer space. For this reason, UAV-App and MR-App implement a synchronization mechanism based on Presentation Time Stamp (PTS) on the transmission and reception modules.

The synchronization process implemented is based on time video processing by the codification and detection modules. On the first step of this process, the video is stored on a UAV-App by a codification module only when a connection module has established that the application can receive data from the drone. Then, the H264 YUV240 video format caught by a codification module is downsized and converted to a four-colour channels video format to fit the detection module methods. Here, a classification method uses a pre-trained model for object (or human) detection that is stored on the knowledge module.

V. OBJECT DETECTION APPROACHES

The proposed architecture could use traditional and deep learning detection methods, and both were tested in this work. Each approach has their own advantages and disadvantages.

A. Traditional methods

Some of the most recommended ([4], [14] and others) traditional methods for human detection are Support Vector Machines (SVM) and Adaptive Boosting (AdaBoost). These are fed by features obtained with Histograms Oriented Gradients (HOG), Haar and others.

B. Deep Learning models

Regarding deep learning models, there are several algorithms that can be used to train real-time detection models as, for example, Faster Region Convolutional Neural Networks (Faster R-CNN) [15], Single Shot Detector (SSD) [16], and all the versions of You Only Look Once (YOLO) [17] [18] [19].

C. Training and evaluation

The training process is preceded by the preparation step, which has three main phases. The first one is where images are collected (dataset). In the second phase, object instances are labeled. In the third step, the dataset is split into training data and test data. In this last step, split combinations are usually used to increase the model knowledge.

The model is trained during some number of iterations. On each iteration, the model's output for all dataset object instances is computed and the number of true-positives (TP), true negatives (TN), false positives (FP) and false negatives (FN) are counted. In the end, error metrics are computed using

equations (1), (2) and (3) or using the Area Under a Curve (AUC) of the Precision and Recall graph.

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (1)$$

$$Precision = \frac{TP}{TP+FP} \quad (2)$$

$$Recall = \frac{TP}{TP+FN} \quad (3)$$

In deep learning approaches, the Intersection Over Union (IoU) metric is used to determinate when a classification is accepted. The IoU is a division of the intersection versus the union of a Ground-Truth bounding box and a Predicted bounding box as illustrated in Figure 5 for a face detection sample. This value indicates the accuracy detection of a dataset and is used to establish correct detections.

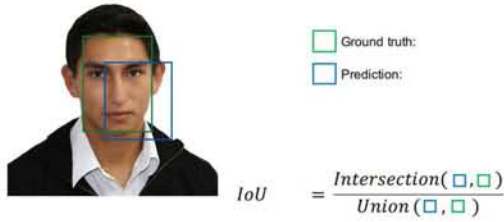


Figure 5. IoU metric

Works using deep learning based approaches, such as [16], [17], [18] and [19], also use a Precision and Recall variation called mean Average Precision (mAP) metric. This metric calculates the detection error with multiple object classes. To obtain the mAP value, an Average Precisions (AP) per object class using the same interpolation is required. Also, it is possible to use multiple IoUs to specify more than one umbral, creating the mAP@ notation. The @ character specifies an initial umbral and the step between the next ones. For example, mAP [0.5, 0.05, 0.95] means that the model is tested with an umbral greater-than or equals to 50%. The Average Precision is computed as follows:

$$AP = \sum_{r=0}^1 (r_{n+1} - r_n) P_{interp}(r_{n+1}) \quad (4)$$

$$P_{interp}(r_{n+1}) = \max_{\tilde{r}: \tilde{r} \geq r_{n+1}} p(\tilde{r}) \quad (5)$$

where r is the recall, $p(\tilde{r})$ is the measured precision at recall $\tilde{r}$ and $P_{interp}(r_{n+1})$ is the maximum precision value that is greater-than or equal to r_{n+1} .

VI. REAL-TIME HUMAN DETECTION

Search and rescue applications present streaming video constraints to display images in real-time. We mention the "real-time" term to refer a predictable execution time of processing image used by the mechanism based on PTS. However, the streaming video also needs to be supported by an object classification model with good accuracy values. Usually, these two constraints depend on datasets features,

training models, image quality and processing unit, topics that are exposed in the next items.

A. Image Quality

The drone cameras provided us 40 frames per second (fps) with an image resolution of 1280x720 pixels, but the resolution and fps were downsized to 2 fps and 320x180 pixels to provide a streaming video with fast image processing. These values were obtained making video transmission tests that evaluate fps with downsize percentage, leading to twenty minutes of video transmission. Obtaining twenty minutes of successfully video transmission is enough because it is the time supported by a flight mission.

In Figure 6 and in Table 2, 14 transmissions without wireless disconnects are illustrated. The sixth, seventh and eighth transmissions obtained a successful video transmission. We can see that transmissions with greater-than 2 fps or better-quality images could obtain fewer transmission times because channel disconnection happens, making reconnection necessary and losing sequences of images while disconnected.

TABLE II. FPS AND QUALITY TRANSMISSION TEST

Transmission #	Minutes	Fps	Quality
1	04:19	1.7142857	27%
2	08:10	1.7142857	25%
3	01:28	3	32%
4	03:11	3	30%
5	15:27	2	32%
6	20:00	2	23%
7	20:00	2	24%
8	20:00	2	25%
9	00:21	2	38%
10	00:50	2	37%
11	00:54	2	39%
12	01:00	2	37%
13	01:16	1.875	33%
14	00:56	1.875	34%

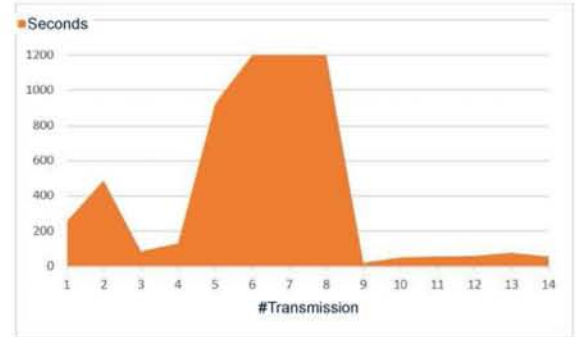


Figure 6. Time (seconds) per transmission

B. Datasets features

The main problems found in, [3], [4], [5] and [8] on datasets for human recognition with drones are related with camera movements (roll, pitch, yaw), capture distance and scene illumination. Using optical and thermal cameras, we could improve the detections in low-light scenarios or in camouflaging foliage scenarios, obtaining fewer detection errors. Nevertheless, in our case, we just deal with capture

distance and camera movements because we do not have thermal cameras installed.

The camera movements and capture distance could be solved using single class datasets such as Generalized Multi-View Real Training GMVRTv1 [3], GMVRTv2 [3], University of Central Florida-Aerial camera, Rooftop camera and Ground camera UCF-ARG [20], as well as multi-class datasets such as Microsoft Common Objects in Context MS-COCO [21], Visual Object Classes PASCAL-VOC [22], ImageNet [23], SUN [24], MobileNet [25] and others.

There are datasets thought for one object class identification (e.g. is there a person in the image?) and datasets thought for the detection of more than one object class, humans being one of them. In the first case, the dataset contains images with and without human instances. Usually, these images are called positives and negatives, respectively. On the other hand, images in multi-class datasets may contain several object instances.

C. Detection models

In [4], it was found that deep learning methods provide better accuracy values than traditional methods, but that the first ones require more execution time. We reached the same conclusion in our experiments because in the first steps of this work, human detection was implemented using the most popular library for traditional methods called Open Computer Vision (OpenCV), with poor accuracy. Then, human detection was improved using deep learning methods over TensorFlow Object Detection Application Program Interface (TensorFlow OB API), but with more processing time per frame.

At the beginning of this project, we tested OpenCV with Haar-like features and the trained model of [26] that provides an average of 90% detection rate. Then, the detection improved using HOG+SVG with the trained models of [27], which provides precision of $\sim 99.4\%$ and recall of $\sim 93.87\%$. However, the datasets [27] and [26] are not focused on detecting people with drones and lead to a significant number of false-positives as illustrated in Figure 7.



Figure 7. Detection with Haar-like features [27] and HOG+SVG [26]

Training a new model using the datasets proposed by [3], [4], [5] or [8] requires significant resources regarding training time, features definition, splitting, hardware with GPU or faster CPU. To solve this problem, we used deep learning methods such as SSD and YOLO, which can be used with TensorFlow OD API and datasets thought for multi-call detection.

The Pre-trained model *ssd_mobilenet_v1_coco* presents acceptable metric values according to mAP and speed presented in Table 3. On the other hand, *tiny-yolo-voc-graph*

[28] is able to recognize the presence of a person with $\sim 0.94\%$ probability.

TABLE III. MAP AND SPEED OF TENSORFLOW MODELS [29]

Model Name	Speed (ms)	mAP	Dataset with COCO
<i>ssd_mobilenet_v1_coco</i>	30	21%	MobileNets
<i>tiny-yolo-voc-graph</i>	Probability $\sim 0.94\%$		PASCAL VOC

Human detection was considerably improved using the faster model *ssd_mobilenet_v1_coco* presented in Figure 8, however, human recognition constraints over angle movements exist yet. In [30] it is shown that transfer learning processes, such as fine-tuning, a model can improve accuracy percentages in specific classes. Also, this theory was demonstrated by [31] when training a MS COCO dataset using a computer with a faster processing unit. For these reasons, we applied a fine-tuning process of deep learning models using traditional datasets to obtain better detections.



Figure 8. Human Detection with *ssd_mobilenet_v1_coco*

D. Processing Units

The type of processing unit could be the main factor that affects the data processing time to offer video in real-time. In [4], it is mentioned that laptops usually have more computational resources than mobile devices to process the data, for example, a faster Central Processing Unit (CPU) or a Graphics Processing Unit (GPU).

We used devices without GPUs for human recognition, but we also used a virtual machine with GPU to transfer knowledge from an object detection model to a human detection model. In [32] the authors found that, unsurprisingly GPUs significantly outperform CPUs machines on training object detection models with TensorFlow.

VII. TRANSFERRING KNOWLEDGE TO A NEW MODEL

Transfer learning focuses on storing knowledge gained while solving one problem and applying it to a different but related problem [33]. Fine-tune an object detection model is a transfer learning process that improves the classification in a subset of categories [31]. This process requires the change of some aspects of the neural network architecture to improve the object classification of a new model. To fine-tune the *ssd_mobilenet_v1_coco* model that is based in COCO dataset, we used TensorFlow tools such in [31].

The preparation process was split in the three phases mentioned in Section V. In the first phase, we choose 440 videos of 10 human actions of the UCF-ARG [20] dataset. This dataset was chosen because it contains images caught in

less than 30.45 meters of altitude. Then, we played all the videos, catching an image per second in jpeg format, which led to a training dataset of 1947 images and a test dataset of 493. In the second phase, illustrated in Figure 9, we labeled the training images in the PASCAL VOC format, which uses XML annotations.

In the third phase, the training and evaluation record files required to the fine-tuning were created. Here, we wrote a pipeline file and hyper-environments to train with GPU resources according to the file structure in Figure 9. The model was trained in a virtual machine with Nvidia CUDA GPUs such as recommended in [34].

Finally, the training process led to a model with an accuracy of 0.89 for human detection.



Figure 9. Labeling persons in UCF-ARG images and file structure

VIII. RESULTS AND FUTURE WORK

This paper presented an architecture for SAR operations that uses a Mixed Reality based interface to provide location awareness and video streaming. The human recognition modules are the most complex task in this work, as well as the training process of the detection model. An object detection model was re-trained using a transfer learning process to detect humans with drones. The model obtained has now a detection accuracy of 0.89 but, as future work, we plan to improve human recognition using thermal cameras.

REFERENCES

- [1] P. Doherty. e. al., "The WITAS Unmanned Aerial Vehicle Project," *14th European Conference on Artificial Intelligence*, Madeira, 2000.
- [2] De Oliveira and M. Wehrmeister, "Using Deep Learning and Low-Cost {RGB} and Thermal Cameras to Detect Pedestrians in Aerial Images Captured by Multirotor UAV," *Sensors*, 2018.
- [3] P. Blondel, A. Potelle, C. Pégard and R. Lozano, "Fast and viewpoint robust human detection for SAR operations," in *2014 IEEE International Symposium on Safety, Security, and Rescue Robotics*, Hokkaido, 2014.
- [4] P. Blondel, A. Potelle, C. Pégard and R. Lozano, "Fast and viewpoint robust human detection for SAR operations," *IEEE, Hokkaido*, 2014.
- [5] P. Blondel, A. Potelle, C. Pégard and R. Lozano, "Human detection in uncluttered environments: From ground to UAV view," *IEEE, Singapore*, 2014.
- [6] X. Ji, X. Xiang and T. Hu, "Data-driven augmented reality display and operations for UAV ground stations," *IEEE, Chongqing*, 2017.
- [7] M. Prexl, K. Struebig, J. Harder and A. Hoehn, "User studies of a head-mounted display for search and rescue teleoperation of UAVs via satellite link," *IEEE, Big Sky*, 2017.
- [8] P. Doherty and P. Rudol, "A UAV Search and Rescue Scenario with Human Body Detection and Geolocalization," *Springer, Berlin*, 2007.
- [9] M. Erdelj. e. al., "Help from the Sky: Leveraging UAVs for Disaster Management," *IEEE*, 2017.
- [10] P. Molina. e. al., "Searching lost people with UAVs: The system and results of the Close-Search project," *Infoscience, Melbourne*, 2012.
- [11] J. Montenegro. e. al., "A Surveillance Task for a UAV in a Natural Disaster Scenario," *IEEE, Rio de Janeiro*, 2012.
- [12] R. Krengkamjornkit and M. Simic, "Human Body Detection in Search and Rescue Operation Conducted by Unmanned Aerial Vehicles," *Advanced Materials Research, Switzerland*, 2013.
- [13] A. Iglesias, "Las 20 compañías que dominan el mercado de los drones," 2017. [Online]. Available: <https://www.ticbeat.com/tecnologias/las-20-companias-que-dominan-el-mercado-de-los-drones/>. [Accessed 27 1 2019].
- [14] Y. Wang, J. Xing, X. Luo and J. Zhang, "Pedestrian Detection Using Coarse-to-Fine Method with Haar-Like and Shapelet Features," *IEEE, Ningbo*, 2010.
- [15] Ren Shaoqing. e. al., "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," *IEEE*, 2016.
- [16] Liu Wei. e. al., "SSD: Single Shot MultiBox Detector," *Springer*, 2016.
- [17] Redmon Joseph. e. al., "You Only Look Once: Unified, Real-Time Object Detection," *IEEE*, 2016.
- [18] Redmon J. and Farhadi A., "YOLO9000: Better, Faster, Stronger," *IEEE*, 2017.
- [19] Redmon J. and Farhadi A., "YOLOv3: An Incremental Improvement," *CoRR*, 2018.
- [20] Nagendran A., Harper D. and Shah M., "New system performs persistent wide-area aerial surveillance," *SPIE Newsroom*, 2010.
- [21] Lin T. e. al., "Microsoft COCO: Common Objects in Context," in *Cornell University, Springer*, 2015.
- [22] Everingham M. e. al., "The PASCAL Visual Object Classes Challenge: A Retrospective," *Springer*, 2015.
- [23] Deng Jeng. e. atl., "ImageNet: A large-scale hierarchical image database," *IEEE, Miami*, 2009.
- [24] Xiao J. e. al., "SUN Database: Large-scale Scene Recognition from Abbey to Zoo," *IEEE, San Francisco*, 2010.
- [25] Howard Andrew. e. al., "MobileNets: Efficient Convolutional Neural Networks for Mobile Vision," 2017.
- [26] Shanshan Zhang, "Informed Haar-like Features Improve Pedestrian Detection," *IEEE, Columbus*, 2014.
- [27] Vision-ary.net, "BOOST THE WORLD: PEDESTRIAN DETECTION," [Online]. Available: <http://www.vision-ary.net/2015/03/boost-the-world-pedestrian/>. [Accessed 1 10 2018].
- [28] TensorFlow, "Android YOLO with TensorFlow Mobile," TensorFlow, 18 5 2018. [Online]. Available: <https://github.com/szaza/android-yolo-v2>. [Accessed 2 2 2019].
- [29] TensorFlow, "Tensorflow detection model zoo," [Online]. Available: https://github.com/tensorflow/models/blob/master/research/object_detection/g3doc/detection_model_zoo.md. [Accessed 2 2 2019].
- [30] Girshick R., Donahue J., Darrell T. and Malik J., "Rich feature hierarchies for accurate object detection and semantic segmentation," *IEEE*, 2014.
- [31] Sonntag Daniel, "Fine-tuning deep CNN models on specific MS COCO categories," *ResearchGate*, 2017.
- [32] Lawrence John. e. al., "Comparing TensorFlow Deep Learning Performance Using CPUs, GPUs, Local PCs and Cloud," *Semantic Scholar, New York*, 2017.
- [33] West J., Ventura D. and Warnick S., "A Theoretical Foundation for Inductive Transfer," *Spring*, 2007.
- [34] Lawrence John. e. al., "Comparing TensorFlow Deep Learning Performance Using CPUs, GPUs, Local PCs and Cloud," *Semantic Scholar, New York*, 2017.