Scientific Research Publishing

International Journal of Intelligence Science, 2020, 10, 41-64
https://www.scirp.org/journal/ijis
ISSN Online: 2163-0356
ISSN Print: 2163-0283

Vision and Geolocation Data Combination for Precise Human Detection and Tracking in Search and Rescue Operations

Eleftherios Lygouras

Department of Production and Management Engineering, Democritus University of Thrace, Xanthi, Greece
Email: elygoura@pme.duth.gr

How to cite this paper: Lygouras, E. (2020) Vision and Geolocation Data Combination for Precise Human Detection and Tracking in Search and Rescue Operations. *International Journal of Intelligence Science*, **10**, 41-64.
https://doi.org/10.4236/ijis.2020.103004

Received: March 20, 2020
Accepted: May 17, 2020
Published: May 20, 2020

Abstract

In this paper, a study and evaluation of the combination of GPS/GNSS techniques and advanced image processing algorithms for distressed human detection, positioning and tracking, from a fully autonomous Unmanned Aerial Vehicle (UAV)-based rescue support system, are presented. In particular, the issue of human detection both on terrestrial and marine environment under several illumination and background conditions, as the human silhouette in water differs significantly from a terrestrial one, is addressed. A robust approach, including an adaptive distressed human detection algorithm running every N input image frames combined with a much faster tracking algorithm, is proposed. Real time or near-real-time distressed human detection rates achieved, using a single, low cost day/night NIR camera mounted onboard a fully autonomous UAV for Search and Rescue (SAR) operations. Moreover, the generation of our own dataset, for the image processing algorithms training is also presented. Details about both hardware and software configuration as well as the assessment of the proposed approach performance are fully discussed. Last, a comparison of the proposed approach to other human detection methods used in the literature is presented.

Keywords

Distressed Human Detection, Unmanned Aerial Vehicles (Uavs), Search and Rescue (SAR) Operations, Aerial Image Processing, Image Processing Algorithms

1. Introduction

Aerial image-based object detection is a very active research topic within the field of computer vision owed to the florescence of UAVs technology. Aerial

platforms are suitable for creating a dynamic visual data collection as they are capable of covering large areas and instantly providing clearer and wider lines of sight over uneven and hazardous terrain. Thus, Search and Rescue (SAR) missions can greatly benefit from aerial robots equipped with intelligent vision systems, as human survival chances are negatively correlated with detection time. The most common detecting object in image processing is human silhouette due to the variety of application domains in which forms an integral part. In spite of human's capability to detect objects in his field of view, this is a particularly complex and challenging task for computers, as human's appearance within the observed region significantly changes due to its dependence on illumination conditions, background, poses variations, shape deformation, camera's selection, etc. Numerous techniques for human detection and classification, action recognition and tracking have been mentioned in the literature, including Histogram of Oriented Gradients (HOG) descriptors [1], background subtraction [2] [3] [4], foreground segmentation [5], optical flow [6], adaptive background subtraction [7], to name a few. The superior performance of such techniques has led to an abundance of research including several types of Convolutional Neural Networks (CNNs), such as Region CNNs (R-CNN), Temporal Convolutional Networks (T-CNNs) [8], 3D CNNs or even deeper models. Developments and implementations of deep learning techniques have been widely adopted in computer vision field as the de facto standard approach. Yet, limitations and drawbacks still exist, as these approaches are capable of achieving excellent results on various detection fields. However, these techniques, compared to other special-purpose detectors, are more computationally demanding and slower. Furthermore, such techniques require huge amount of dataset in order to be successfully trained as well as high-end power consuming Graphics Processing Units (GPUs), heavy enough to be on-board a lightweight and low-cost UAV.

Almost all types of cameras have been proposed and evaluated for the human detection accuracy enhancement. The most widespread video devices are visual cameras, generating grayscale or RGB images [9] [10]. Some drawbacks related to the utilization of such devices under several conditions are: the objects' visibility which depends on energy sources (e.g. artificial lighting, sunlight etc.), the non-uniform illumination, shadows, low external light during the evening and night, colours balance, direction, etc. Infrared light vision cameras have been used as an alternative for the detection of human bodies because of their distinguished thermal signature. Abundance of research for human detection using both near-infrared (NIR) [11] and far-infrared (FIR) cameras or thermal cameras [5] [12], has been conducted. Despite the fact that using FIR cameras can be beneficial, their performance can grievously be affected by factors, such as low image resolution, low contrast, thermal images noise, as well as the background's high temperature during daytime. Moreover, thermal cameras resolution is lower compared to visible light cameras and their cost is extremely high, making them difficult to be adopted in low-cost vision-based surveillance systems. Vi-

sion/NIR cameras are not presenting the above issues. Their performance is superior during the day and night. Some drawbacks of the NIR cameras include the limitations in terms of adjusting the intensity and angle of the additional NIR illuminator according to its distance from an object. The power of artificial lighting is capable of detecting humans in the range of 20 - 80 m. In order to avoid the bright object's saturation in the captured image, the artificial lighting power has to be adaptively adjusted in case the object is closer than 20 m. However this is automatically done in most commercially available cameras.

Aerial image processing has been an extremely active area in recent years due to its increased data availability and its feasibility in various scientific fields [13] [14]. UAVs are suitable for image capturing and processing purposes owing to their superior capability of providing a profusion of precious visual information. Thus, SAR operations can greatly benefit from such aerial vehicles as their efficacy is abundantly depended on the instantaneous and precise detection of humans enmeshed in events threatening their physical integrity [15] [16] [17] [18] [19]. The contribution of UAVs in such missions is indisputable as they are capable of:

- reducing the critical response time.
- mapping areas where an emergency event occurs, directly detecting potential victims and providing assistance to them until the rescuers arrival.
- providing access into hazardous terrains.
- providing crisis event aerial visibility event's overview for the rescue workers' accurate situational awareness.

Notable research has been conducted towards vision-based victim detection in SAR missions. A review of human body detection methodologies conducted by UAVs for SAR applications is presented in [20]. Several methods and algorithms for the implementation of human detection frameworks are highlighted. Assessment and potential improvements of the aforementioned algorithms are also discussed.

Recent research emphasize on the suitability of deep learning techniques adoption for assisting SAR missions supported by UAVs [21] [22]. Such approaches involve autonomous navigation of aerial vehicles [23], exploration of various unknown cluttered environments [24] or rescue missions conduction in indoor environments for human presence recognition [25].

Although plenty of research has been conducted towards detection and tracking in coastal environment-based on a large amount of available dataset-the corresponding research and dataset for humans in open water detection is limited. Furthermore, there has been no relevant substantive research concerning the adoption of computer vision algorithms for the conduction of SAR operations supported by UAVs during both day and night-time hours under several backgrounds. Hence, the novelty of this research banks on the implementation and the evaluation of GPS/GNSS techniques combined with advanced image processing algorithms for the enhancement of SAR missions' efficacy in both

coastal and marine environments under several illumination conditions using a day/night low cost vision/IR. Most of the detection and tracking algorithms are mainly based on Python [26], Open Source Computer Vision Library (OpenCV) [27] and Tensorflow [28] running in real time even at lower fps rates. Especially in our case studies, rates down to 4 fps are acceptable for the sought application. Moreover, in order to generate a sufficient size dataset of humans in peril under several use case scenarios, the majority of the training dataset are images cropped from videos captured by the rescue system's UAV flying at several altitudes and on a variety of spots such as mountains, river banks, and open water.

The majority of the existing research has mainly focused on human's detection methods during daylight hours, as image capturing under low-light intensity or total darkness is troublesome. In this article, we study and evaluate the efficacy and the feasibility of open source libraries, software routines and advanced image processing algorithms, on a fully autonomous UAV-based rescue support system. The main objective of this research is summarized as follows:

- Creating two different datasets for human in peril detection from an aerial rescue support system. The first one contains human in land high resolution images, while the second one contains human in open waters high resolution images. Both datasets contain images during both daylight and nighttime hours, captured from the fully autonomous rescue UAV.
- Testing and proposing the use of open access freely available custom trained and pre-trained models for the precise detection and tracking of human in peril, under several conditions (illumination, human poses, backgrounds, etc.).
- Proposing a hybrid approach by combining human detection algorithm running every N frames of the input image, combined with an adequately fast and accurate tracking algorithm. Thus, real time or near-real-time human detection/tracking is achieved.
- Studying and evaluating the distressed human detection challenge under two use cases; human in land and human in water detection.
- Proposing the use of a single, low cost day/night camera for human detection during all illumination conditions.

The remainder of this paper is structured as follows: In Section 2 the methodology for the distressed human detection is fully discussed. Details about the implementation of the precise human detector and tracker are presented in detail in Section 3. Section 4 presents experimental results of human in land detection and tracking while Section 5 presents human in open water detection ones. The paper is concluded in Section 6.

2. Detection Approach

Combined GPS/Vision Human Detection System Description

The proposed aerial rescue support system's architecture and functionality have been presented in detail in [15]. The autonomous rescue system's aerial vehicle

is capable of conducting search and rescue missions from takeoff to landing, without any human intervention, guided solely by data provided by humans' in peril wearable apparatus. The issue of the rescue UAV precisely positioning upon the target point has been extensively studied and several approaches have been tested and evaluated. Considering the possibility that the supervised person (SP)[1] position is dynamically changing due to several reasons, the integration of a vision system in the loop is apparent. Figure 1 depicts the whole rescue procedure flowchart. Once a SAR mission is activated to the Ground Control System (GCS), the GCS automatically designs UAVs flight path and directs it to

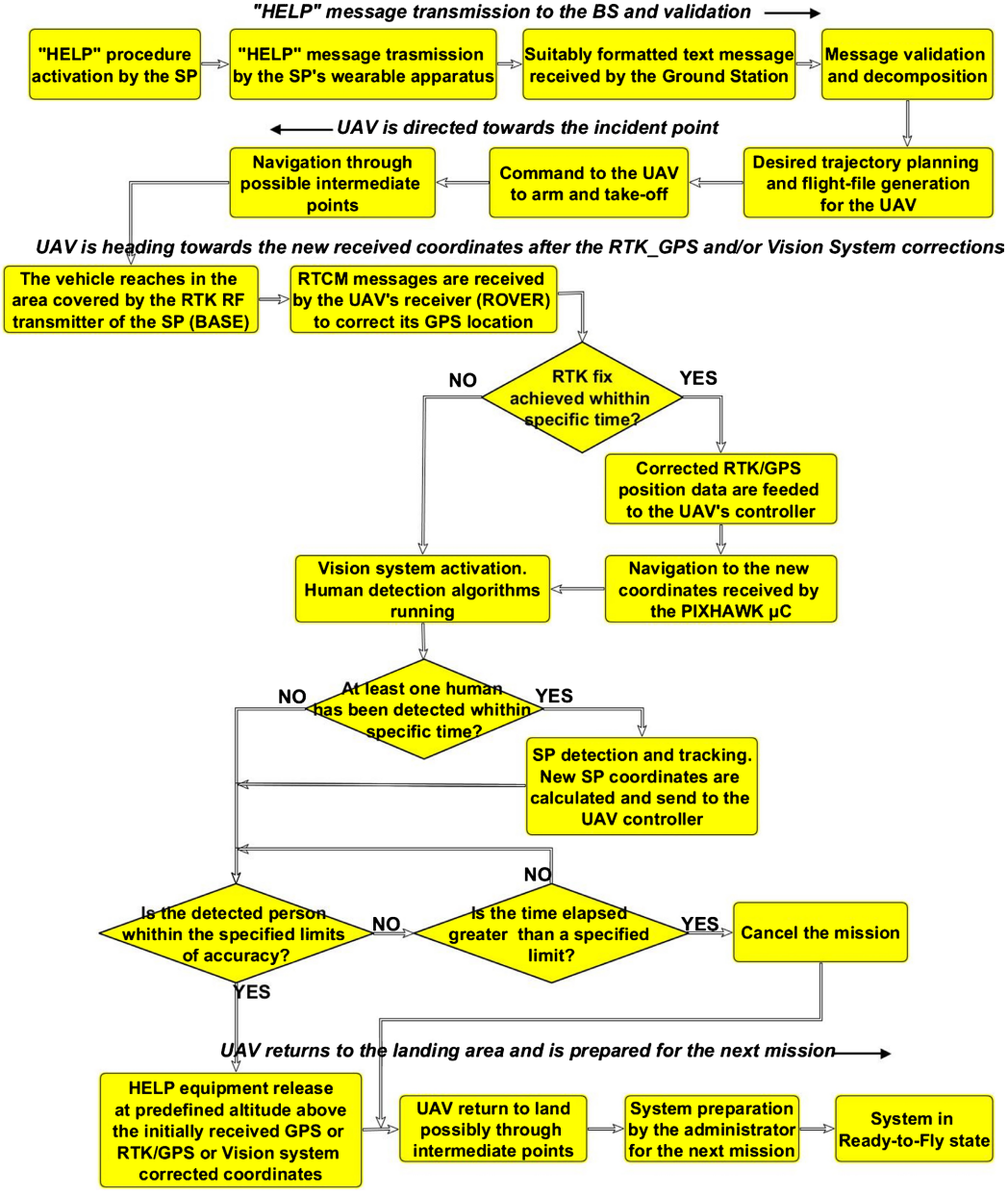


Figure 1. The fully autonomous UAV rescue mission flowchart combining RTK-GPS techniques and image processing algorithms.

[1]Supervised Person: a person to be benefited by the aerial rescue support system.

the incident area. The UAV's autonomous navigation to the incident area consists of the following three distinct phases:

1) The UAV heads to the incident area based on the GPS/GNSS coordinates received from the "HELP" text message of the SP's wearable equipment.

2) When the vehicle is within the range of the RTK data link radio transmitter of the SP's wearable equipment and provided that an adequate amount of satellites are visible from both GPS/GNSS receivers (on base and rover GPS), Real Time Kinematics (RTK) GPS fix[2] is achieved and the vehicle is moving towards the SP with improved accuracy compared to a simple GPS/GNSS receiver. Thus the UAV reaches upon the SP at a determined height and depending on the system accuracy is located inside a circle with radius determined by the RTK-GPS accuracy. If not RTK-GPS fix has been achieved, the vehicle continues moving towards the target based on its simple GPS receiver information received by the last communication with the ground-station.

3) At this stage, the UAV descends over the SP and the task of analyzing the captured video is conducted through an automated algorithm in real time, by identifying human bodies in the vision/IR camera video frame. Provided that at least one human has been detected, the calculation of its pixel coordinates, locates where human bodies have been found, place the detected persons in rectangular boxes and calculate their centroids. If more than one human are detected, the aerial vehicle heads towards the centroid exhibiting the minimum Euclidian distance from the SP's RTK-GPS received coordinates. This information, combined with its geographic coordinates and the UAV's speed in the horizontal plane X and Y axes, are controlled in order the vehicle to position itself exactly upon the SP and to follow him/her in case of moving and to release its help equipment at a pre-specified altitude. If the human detection algorithm fails, the vehicle moves towards the target based on the RTK-GPS coordinates or alternatively on its simple GPS/GNSS receiver information.

This paper focuses in the third phase of the SP approach, since the other two have been described in our prior research.

Videos captured by aerial vehicles are pretty dissimilar to the ones acquired on the ground in respect both of pose and content. Therefore, the use of conventional processing techniques may not apply in all cases. Thus, the following considerations should be examined at the stage of the image processing algorithm design. Firstly, the distance between the target point and the camera is greater than the standard "ground" techniques. Moreover, the human silhouette in terrestrial environment is different from an aquatic environment one. In particular, human in dry land should be totally visible, while a human silhouette in aquatic environment is most likely to be partially visible partially or mostly the upper body. Partial occlusions should also be considered in both these two cases. Moreover, should the vehicle to be operational during all day and night hours, the human in peril detection algorithm must be efficient under several lighting conditions.

[2]The time-based process required for a GPS to acquire enough visible satellite signals and data to provide accurate navigation.

The main characteristic of day/night cameras, featuring an IR-cut filter, is that they can operate effectively during all times of the day, making them suitable for rescue UAVs. This is due to their capacity for keeping the disturbing infrared light out of the image sensor during the day light. When the light falls below a certain level, the filter automatically retracts and the infrared light hits directly the image sensor. At the same time, the camera switches to black and white mode. Color CMOS cameras are using infrared cut-off filters to produce HD colour images during the day. The infrared light transmission is blocked by the reflective IR filter and enables the passing of the visible light to allow correct reproduction of colours. When the camera is in night mode, the IR-cut filter is removed. This allows the camera's light sensitivity to reach down to 0.001 lux or lower, providing an excellent B/W video output. An IR illuminator that provides near-infrared (NIR) light is also included in the day and night camera enhancing the camera's ability to provide high-quality video under several lighting conditions.

The camera used in this research is a very low cost true day/night CMOS camera and some of its technical specifications are presented in Table 1.

Vision system's architecture is depicted in Figure 2. The UAV is equipped with a vision/near IR (NIR) camera that can operate during both day and night illumination conditions. The captured video is sent through a transmitter/receiver to the main GCS's main controlling computer. The received video is being automatically processed on this computer (equipped with a GPU), which is also responsible for the aerial vehicle's control. The image processing algorithm returns the detected in each frame human's coordinates and the computer transmits back to the UAV the computed in each axis UAV velocities in order to locate itself precisely upon the SP. This architecture presents numerous advantages, as it provides more available computational power and memory compared to an onboard system, as well as less weight for the UAVs payload capability. Thus, human detection process is performed on ground; to the GCS controlling computer. Since the rescue apparatus depends on the kind of application the system is used for (*i.e.* marine or coastal environment), the human detection and

Table 1. Day/Night IR camera specifications.

Technical Features	Description
Back Light Compensation	Yes
Auto Gain Control	Yes
White Balance	Yes
Sensitivity Lux	0 ~ 0.01
Day & Night	IR cut filter with auto switch
IR range	40 m
Image Sensor	CMOS
Effective Pixel	1296 × 732

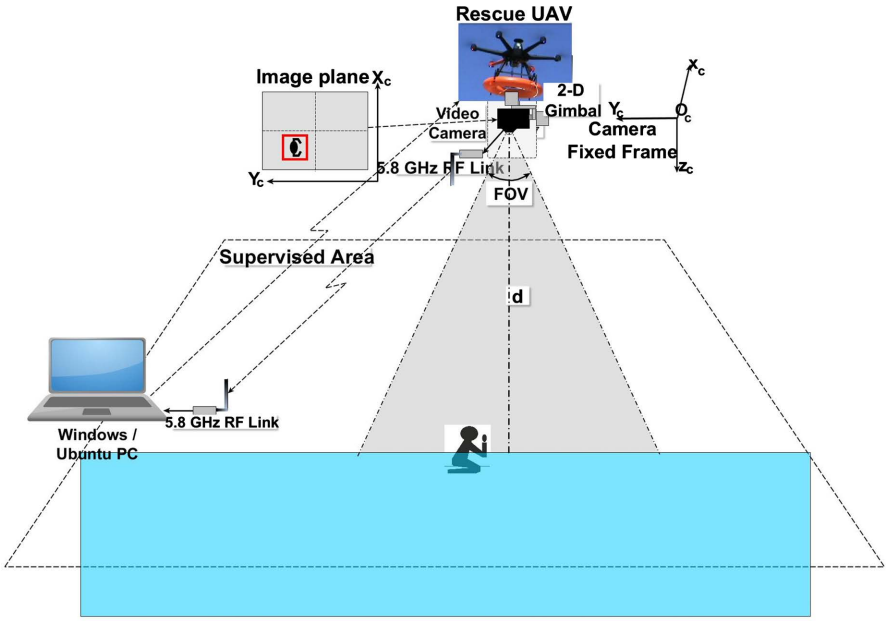


Figure 2. Vision system's architecture for human detection from the rescue UAV.

tracking software can be separated depending on the background of the use case scenario.

3. Implementation of Real-Time Human Detector/Tracker from UAV Video-Feed

3.1. Human Detection and Tracking Algorithms

In order to implement the UAV-based rescue apparatus release based on image processing, an effective human detection, localization and tracking algorithm is apparent. The task of detection involves the identification of human presence and the drawing of a rectangular boundary surrounding the human (*i.e.* human localization). Combining the existing UAV positioning control system, with a Python object detection algorithm which can detect the "Human" class, a vision-based human detection system is developed.

OpenCV is an open source software library, providing a common infrastructure for image processing purposes and applications. It includes thousands of traditional and state-of-the-art optimized algorithms, supports a set of algorithms related to a wide variety of programming languages like C++, Python, Java and is available for several operating systems such as including Windows, Linux, OS X, Android, iOS etc. Examples of human detection algorithms that could be used include color thresholding + contour extraction, HOG + Linear SVM, Haar cascades, as well as deep learning-based detectors such as Single Shot Detectors (SSDs), Faster R-CNNs, and YOLO. Yet, these detectors are in general very computationally expensive, especially when they run on a single CPU.

There are a lot of online available open-source pre-trained human detection models, but there is also the possibility to train our own custom image prediction model. There is a variety of custom image prediction model training me-

thods using state-of-the-art architectures such as InceptionV3, ResNet50, SqueezeNet, DenseNet etc. This allows us to train our own model on any set of images that corresponds to any type of objects/persons. Model's training is generally an intensive and time-consuming compute task and the users are commonly performing this task using a computer with a NVIDIA GPU combined with Tensorflow. Performing model training on CPU may take hours or days. In this research, both Faster R-CNN and SSD-Resnet are employed to detect humans in peril, in cooperation with several tracking algorithms. The detectors are chosen as typical state-of-the-art detectors.

3.2. Adaptively Changing Detection/Tracking Algorithm

A straightforward way to implement human detection and tracking in both coastal and marine environment is the utilization of algorithms mainly in Python and OpenCV running in real time, without this being a prerequisite, as slower algorithms with adequate detection results (down to 4 fps) are acceptable for our application. During the detection phase, the human tracker is apparent in order to:

- Detect if new humans have entered the scene: accepting the input (x, y)-image coordinates of human position in each frame and assigning an individual ID to each of them,
- Track the human while he/she moves out of the video frame's boundaries limits around the frames, predicting the human position in the next frame based on various frames' attributes such as gradient, optical flow, etc.,
- Be able to pick up human that the detector has "lost" between frames.

Thus, the detecting algorithm may be adjusted to run once to every N frames and then applying a faster human tracking algorithm in all frames for a more efficient object tracking pipeline. Moreover, in our case, things are commonly simpler than other human detection systems since we expect only one human (or a small number of humans) to be present in the scene, during the final UAV approach phase. It is also significant that the help equipment can be released either over the person asked for the help but also it will be helpful if it is released to another person alongside the SP, within some very small limits of-coarse.

A human detector is typically more computationally demanding and slower, than a human tracking algorithm. Thus, a combination of a human detector with a human tracker could potentially provide a fair compromise between speed and accuracy in human detection. The detection phase runs once every N frames. This number defined by the user based on the system requirements. Next, the tracking algorithm runs until the N-th frame, at which instance the object detector resumes. The entire process is repeating. The hybrid adaptively changing detection/tracking algorithm is as follows:

- When an object (human) is detected, the detector generates: $B_{Det(i)} = [x_i, y_i, w_i, h_i]$ bounding boxes for each human in the scene. A new unique ID is assigned for every detected object, if it cannot be associated with an old object.

The tracking algorithm is initialized for each one of these bounding boxes. The tracker is active for the next at least N frames of the input video sequence. The detector is activated again after N frames and a new output of: $B_{Det(t+1)} = [x_i, y_i, w_i, h_i]$ is generated. The output bounding boxes at $t+1$ is denoted by:

$$B_{Trac}(t+1) = \left[x_j, y_j, w_j, h_j\right] \tag{1}$$

The algorithm computes the Euclidean distance by the formula:

$$D(t+1) = Norm\left(B_{Det}(t+1) - B_{Trac}(t+1)\right) \tag{2}$$

between each pair of existing tracker objects and new detector input objects.

Next, an association of centroids with minimum distances between subsequent frames is accomplished in order to correspond the tracked humans. If the detector fails to detect at least one human at time $t+1$ we continue to run the tracking algorithm and re-run the detector every $N/2$ frames for a predefined maximum number of frames Nmax. If no human is detected it is supposed that no human exists any more in the scene and the algorithm is terminated.

Most common human tracking algorithms include GOTURN, MedianFlow, Kernalized Correlation Filters (KCF), Discriminative Correlation Filters (DCF), MOSSE and CSRT. If a human body is recognized in a frame, the upper left and lower right rectangle's coordinates are calculated, as well as its centroid.

3.3. Datasets and Test Bench for "Human" Class

In order to cover as much as possible use cases scenarios for the detection of distressed humans, including several, backgrounds, poses and lighting conditions, we captured more than 50 videos from the aerial rescue UAV, flying at several altitudes with different angles and views, as well as several conditions (e.g., backgrounds, illumination etc.). Parsing them, our dataset is composed of 32,437 frames, combined with images available online, each one containing at least one human in the frame in a variety of poses instances. A wide range of scales also included, from close-ups to very distant (up to 20 m) views, while image resolutions ranged from 500 × 333 pixels to 1296 × 732. Despite the numerous available datasets for human in land detection, human in marine environments dataset is distinctively restricted. Thus, a development of a specific dataset including humans in open water was apparent.

The implementation and the results of this research were obtained using a laptop with the following specifications: Intel Core i7, 4.6 GHz, RAM 16 GBDDR4 SDRAM, GPU Geforce NVIDIA GTX1050, RAM GDDR5 4096, Windows 10 and Ubuntu 16.04, Open CV.

4. Human Detection and Tracking on Land

In this section, the development of a complete software tool, capable of enabling the autonomous UAV to detect and localise a distressed human in land, while precisely releasing the suitable rescue apparatus is presented. Thus, we have evaluated several state-of-the-art computer vision techniques to demonstrate

that human detection using artificial intelligence and machine learning is feasible and accurate. In particular, we study and evaluate the utilization of two different models for the precise human in land detection implementation; the *faster_rcnn_inception_v2_coco* model combined with the *centroid tracking algorithm* and the *faster_rcnn_resnet*50*_coco* model combined with the *dlib* [29] open source software library for its correlation tracker implementation capability. In both methods the installation of Python with Tensorflow, OpenCV and other required libraries is apparent.

Tensorflow Detection Model Zoo consists of 16 Object detection models pre-trained on COCO Dataset [30]. The top 12 of these models provide "boxes" as output and they are compatible with the code used in this paper. These models are capable of detecting 80 types of objects including humans. Thus, we studied and evaluated these models for human in peril detection and each one of these eight available object tracking algorithms (*i.e.* OpenCV 3.4) are suitable for our application. Taking into serious account their pros and cons, we concluded that CSRT tracker is apparent for higher object tracking accuracy, even with slower FPS throughput, KCF tracker when faster FPS throughput is required with slightly lower object tracking accuracy and MOSSE Tracker when low speed is acceptable. The faster r_cnn object detection algorithm combines two models: a deep CNN for region proposing and the fast R-CNN detector that processes and classifies these regions into object categories. The procedure for the implementation of our precise human detector/tracker using Artificial Intelligence (AI) available tools, is as follows:

1) Installing OpenCV, NumPy, dlib, imutils, Tensorflow and *faster_rcnn_inception_v2_coco* model from Tensorflow Detection Model Zoo.

2) Running the specific Python code, restricting the detected classes for human detection (class 1) based on the above model.

3) Localizing the detected human in the video captured from the wireless link of UAV's on-board camera.

4) Extracting the (X_c, Y_c) coordinates in camera coordinate system.

5) Implementing a human tracking algorithm using Open CV that might operate in cooperation with the human detection algorithm. The human tracking algorithm may not run on any input frame but every N frames in order to reduce the computational time required and also to prevent human occlusions.

6) Calculating the UAV's velocities in X and Y axes and transmit them to its controller, until the UAV locate itself exactly upon the tracked distressed person.

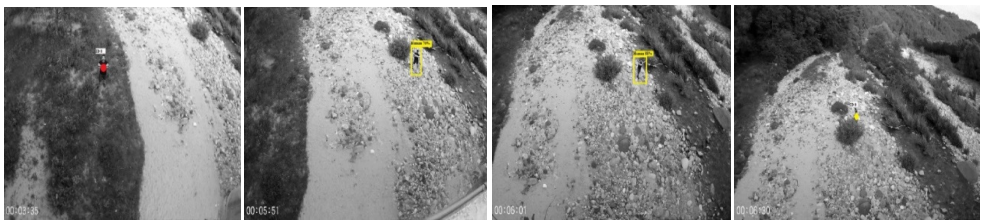
An algorithm which combines high accuracy with faster FPS throughput is the centroid tracking algorithm. This algorithm is an optimal option among numerous other algorithms suitable for human tracking. It is based on passing a set of bounding boxes (X, Y) coordinates for each detected object in each frame. These bounding boxes are generated given that they are computed in each frame of the captured video.

Although the performance of the tracking algorithm was sufficient, some li-

mitations and drawbacks have been encountered. For instance, object detection algorithm is required to run on each frame of the input video. This makes the centroid tracker algorithm more computationally demanding. The second drawback is related to the fact that the algorithm may swap the object's ID, if an overlap of two or more objects is detected, to the point where there is an intersection of their centroids. However, the overlapping object issue is not specific to centroid tracking method as it occurs in even more advanced tracking algorithms. Experimental results from implementation of the *faster_rcnn_inception_v2_coco* detection model and centroid tracking algorithm are depicted in Figure 3, including a distressed human on a rocky field (a) and a river bank (b) during daylight, and a human in peril in park during night-time hours (c). Video footage has been captured from several UAV's flight altitudes, different angles and views and several human poses adoption.

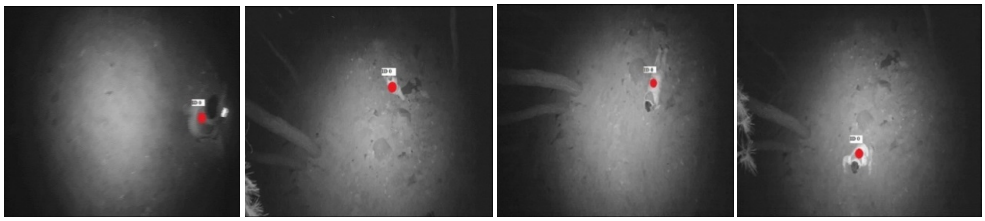
In order the human tracking algorithms to process each frame faster, the input frame is resized to 512 × 512 pixels, since the less data for process, the faster our human tracking pipeline runs. Next, a conversion to grey-scale and application of a Gaussian blur is apparent in order to remove high frequency noise as well as to focus on the "structural" objects of the image.

Faster_rcnn and *ssd_resnet* detectors can detect numerous classes. However, in this research we focus on the successful detection and classification of humans



(a)

(b)



(c)

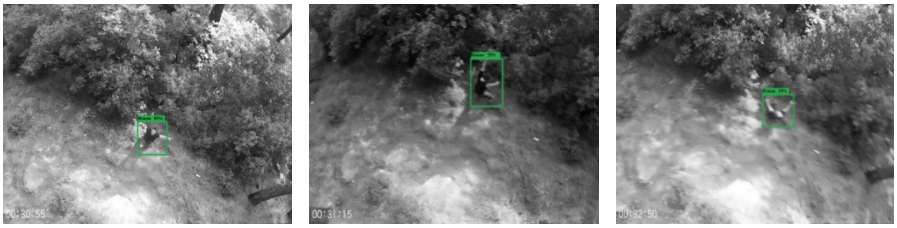
Figure 3. Experimental results from testing the *faster_rcnn_inception_v2_coco* model with the *centroid tracking algorithm* from UAV supplied video feed for distressed human on a rocky field (a) and a river bank (b) during daylight and human in park during night-time (c).

in the scene excluding other objects. Thus, a frame is classified as true positive (TP) if there is a human in the scene and the detector has successfully detected him; as false positive (FP) if there is not human in the image but the detector erroneously reports a detection; as true negative (TN) if there is not human in the scene and the detector does not reports any detection of a human; and as false negative (FN) if there is a human in the scene but the detector does not returns any detection or erroneously labels the detected human. **Figure 3** illustrates results for all the aforementioned cases. Accuracy is calculated based on the formula:

$$\text{Accuracy} = \left(\text{TP} + \text{TN}\right) / \left(\text{TP} + \text{TN} + \text{FP} + \text{FN}\right) \tag{3}$$

An alternative approach for human detection and tracking is to employ both OpenCV for standard image processing functions along with the deep learning object detector for human (*i.e.* the *faster_rcnn_resnet*50_*coco* model) and *dlib* for implementation of correlation filters as human tracker. Experimental results from faster_rcnn_resnet50_coco model implementation with the dlib human tracking algorithm in land during day-time and night-time hours are presented in **Figure 4**. A detection sample is given for several studied use cases, where the rectangle presents the detector's results and its confidence score above it. Failed detections (FN) are depicted in **Figure 4(a)**. In the first line, the reason is the partial occlusion of the undetected human, while in the next two the detector just fails to detect the human inside a circle. Results include distressed human in cliff (a), (b), in river bank (c), (d), in park (e) and in forest (f) during both

(a)



(b)

(c)

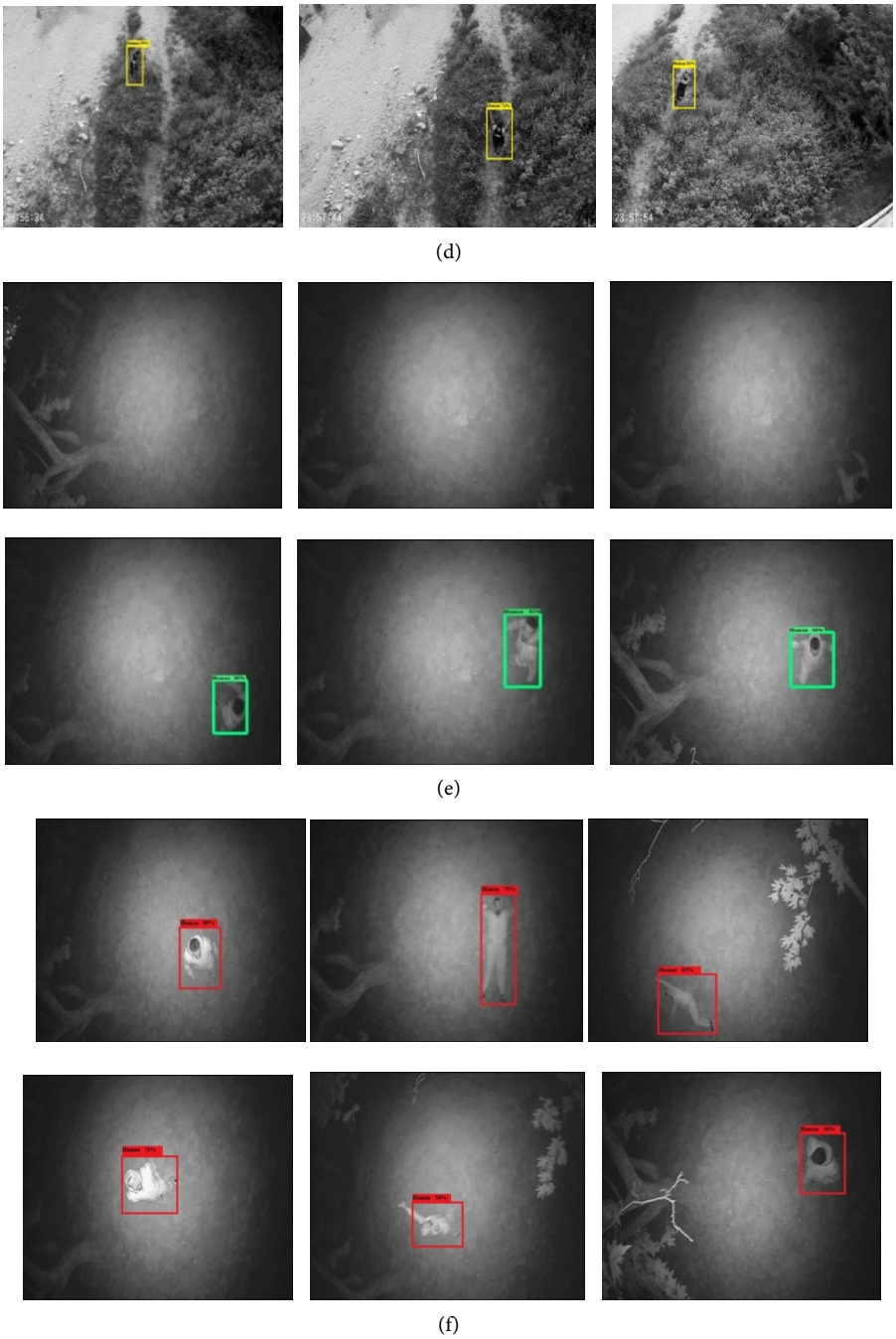


Figure 4. Experimental results using *faster_rcnn_resnet*50_*coco* model and the *dlib* human tracking algorithm in (a), (b) cliff, (c), (d) river bank during day-time and (e) park, (f) forest during night-time.

night-time and day-time hours during night-time. In the first two pictures in (e) there is no human in the scene (TN). In third picture, even the human eye is difficult to discriminate a human in the lower right part, if you ignore its existence (FN). It is more evident in the fourth image (TP). In all other images human in peril is detected and tracked precisely. An overview of the obtained accuracies is presented below.

5. Human Detection and Tracking in Open Water

The objective of this section is the development of a complete software tool that enables the autonomous UAV system to detect and locate distressed humans (e.g. swimmers) in marine environment. The most common challenges for human in marine environment detection are the following: shadows and the reflective regions as background movements could be miss-identified as foreground objects' movements for human detection. Moreover, a very challenging task is the accurate segmentation under poor visibility conditions due to reflections from sunlight and night-time lighting and potential occlusions. Apart from the above unique issues in aquatic environment, there are some other challenges faced in outdoor surveillance such as illumination indensity changes due to ambient lighting, cameras auto-gain issues etc. Fast background updating has to be adapted to such illumination changes. Corruption of the background model may in some cases lead to more segmentation errors on the following frames. Other challenging tasks are the necessity of an algorithm running in real-time and suitable for implementation at low power consumption, low cost, and using available hardware platforms.

This implementation is based on algorithms based mainly in Python, OpenCV and Tensorflow running, if possible, in real or near real time. Thus, we evaluate the performance of the suitable approaches for the detection of swimmers in peril, using artificial intelligence and machine learning algorithms.

There is a collection of pre-trained models on Microsoft COCO [30] dataset, detecting various objects. These models are suitable for out-of-the-box inference and for initializing our models when training on novel datasets. Two modern approaches have been utilized for human in marine environment detection and tracking. Both methods are supported by Python. The first one is based on training our own model. The training was not conducted from scratch, but is based on transfer learning for its efficiency enhancement. The model was trained by employing a fully-trained model (*faster_rcnn_resnet*101_*v*1) and then retraining from the existing weights for the new swimmer classes. The second one is based on using a recently released by Google object detection Application Programming Interface (API) [31]. This API has been trained on Microsoft COCO dataset with different trainable detection models.

The procedure of developing a human detector using AI available tools begins from training our own model. The training may start from scratch, and the procedure may be a long time-consuming process, but training pre-trained models is less time consuming. It includes the following steps:

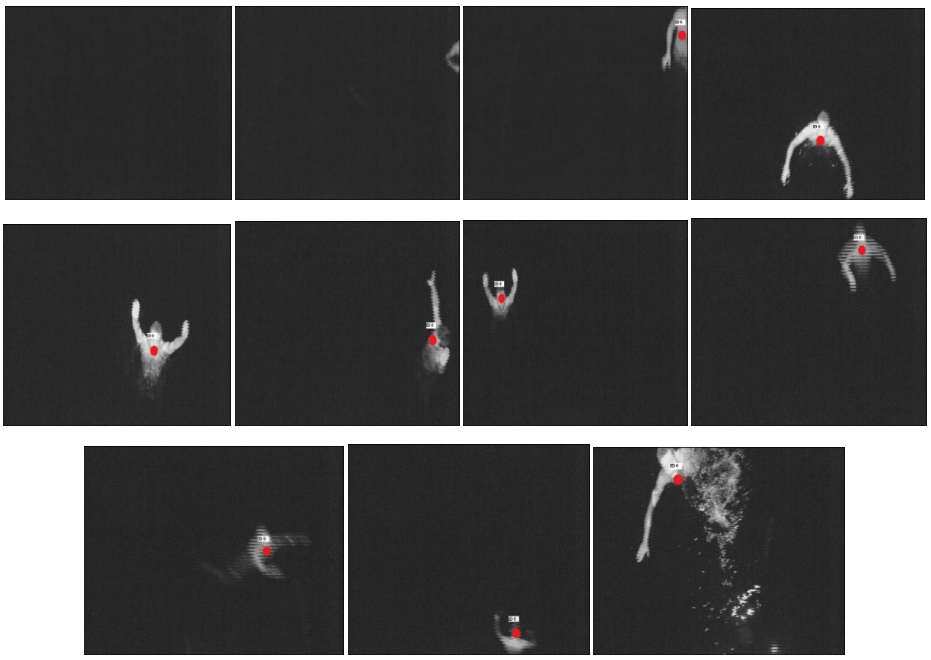
- We initially parse our videos captured under several conditions including humans in open water, in order to obtain an adequate number of frames. Thus, the class "humans" was created using these images for the network's training. Images were selected to show humans up-close and from large distances. We have also included images with variations on their composition, with most images having one human while others having more than one

human. Image quality also varied from high-resolution images to very low-resolution images, and both day and night-time hours. Open source annotation and labeling tools were used for image and video assets during the labelling process; theVisual Object Tagging tool (VOTT) and LabelImg VOTT. In this stage, rectangular boxes are drawn around the swimmers and assigned a label. About 5000 images were totally labelled.

- After images annotation, the exported labels are converted into tensor records, by converting the.*xml* labels into.*csv* format by the *xml_to_csv.py* script, and then parsing the.*csv* file into a suitable.*py* script.
- The next step is the selection of the pre-trained model. There's a trade-off between accuracy and detection speed as the higher the speed the lower the accuracy and vice versa although there are models providing a good trade-off between them. In this paper we use *faster_rcnn_resnet*101_*v*1.
- At this stage the model has to be trained using AI techniques in order to be able to recognize swimmers in open water. The labels, converted to.*csv* file, tensor records, images and label map, along with a pre-trained model and its config file are fed into the training script. This process takes a few hours. However, this time improves as training progresses. Information about the neural network's training progress may be gathered by using Tensorflow's in build tool Tensorboard.
- The python object-detection script is now being activated. After model's training, we can now parse and detect swimmers in the captured video, from the UAV's on board vision/IR camera, with some minor alterations to Tensorflow's object-detection script, to incorporate OpenCV and use its capabilities. This step provides the (X_c, Y_c) coordinates of the detected human in camera's coordinate system.
- The code is used in sequence with a human tracking algorithm also using OpenCV. The human tracking algorithm could also not run on any input frame but every N frames in order the required computational time to be reduced and also to prevent human occlusions.
- Calculate the velocities in X and Y directions for the UAV and transmit them to the UAV controller, until the UAV reaches locates itself upon the tracked person.
- The final step is the selection of a precise tracker to be combined with the human detector. KCF tracker exhibits superior performance for our application. Although the idea of KCF tracker is simple enough, it is one of the recent fastest and top-performing trackers. The key advantage of KCF tracker is that employs the augmentation of negative samples, thus enhancing the discriminative ability of the track-by-detector scheme while exploring the structure of the circular matrix for the high efficiency. As mentioned above, the combination of a detector with a tracker provides for a number of advantages as the reduction of sampling rate and consequently the computational cost.

Figure 5 illustrates experimental results from implementing our *faster_rcnn_resnet*101_*v*1 model for swimmers detection, during both day-time and night-time and under several background conditions. In the first picture in (a) a small part of a swimmer exists (just his hands) (FN) while in the second a human eye can discriminate with difficulty, due to reflections from sunlight, a

(a)



(b)

Figure 5. Experimental results from the implementation of *faster_rcnn_resnet*101_*v*1 model and the *KCF/HOG* tracking algorithm for swimmer detection during day-time (a) and during night-time (b).

swimmer (FN). In the third picture of (a) there are two swimmers in the scene but just the one is detected (TP) while the second is partially occluded and is not detected (FN). In the first picture in (b) there is no swimmer in the scene (TN), while in the second a small part of a swimmer exists (just his hands) (FN). In the rest images the swimmer is detected successfully (TP). The fully trained model has been retrained from the existing weights for the new swimmer classes, providing increased accuracy as it will be described later.

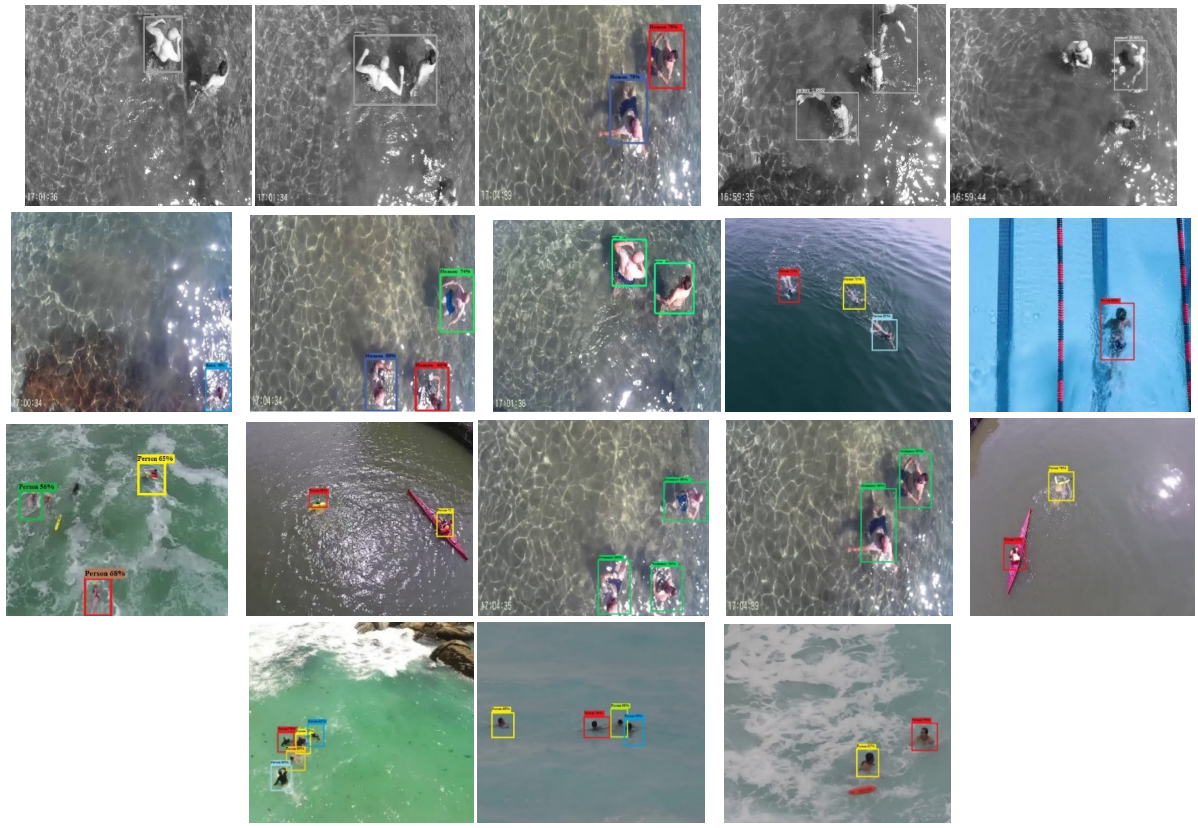
The CNN is capable of accurately detecting and classifying a human object in the image, even if the full contours of the swimmer of interest were obscured by another swimmer. Furthermore, the CNN was able to classify and detect human objects even if they were not fully (partially) shown in the image. For instance, at the fifth image of **Figure 5(a)**, only partial contours of a swimmer were shown and a full contour of the other two swimmers. Images during day-time include one, two and three swimmers in the scene.

In our second approach we are using a recently released by Google object detection API. Firstly, we have to install all the apparent libraries, as pillow, Tensorflow etc. Next, we download the latest version of "*protobuf-python*".*zip* file compatible with our system (Anaconda 3/Spyder, Python 3.6) and clone to the Tensorflow/models folder. The *ssd_resnet*_50_*fpn_coco* model was pretty fast with lower accuracy compared to others methods, but applied to our own images exhibited interesting results. At this point, the whole process is much easier since no model training is needed as we can use directly Google's released object detection API, trained on Microsoft COCO dataset [30].

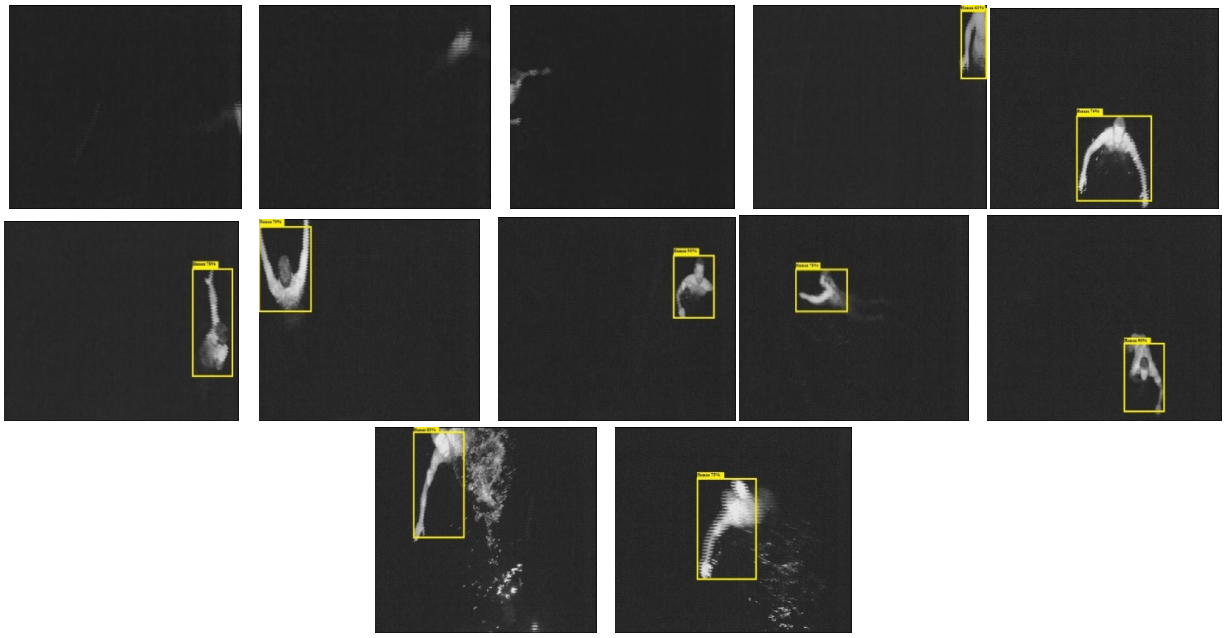
Figure 6 depicts experimental results of human in peril detection and tracking in marine environment, including images available online and ones obtained by the author, under different background conditions. Those are experimental results excluded using Tensorflow *ssd_resnet*_50_*fpn_coco* model and KCF/HOG human tracking algorithm. A rectangle with different color is drawn around each human, if one new is detected. In the first four pictures in (a) one swimmer is not detected, or a rectangle is drawn around two swimmers, however this is not considered as a problem in our application since at least one swimmer, very close to another is at least detected. In the first picture of (b) there is no human in the scene (TN). In the second picture in (b) even a human eye can with difficulty discriminate that there is a swimmer in the scene (FN), while in the third a small part of a swimmer exists (just its head and hands) (FN). All the rest swimmers are detected successfully (TP), even under complete darkness conditions.

Table 2 summarizes the characteristics of the implemented algorithms in this research and a comparison of their mean accuracy and throughput. The results present acceptable detection accuracy for each one of the selected models. In particular, if the algorithm detects at least one human in the scene, even if it contains more than one, the detection is considered to be successful. The accuracy of Faster R-CNN in human shiluette detection outperforms the SSD's one. On the other hand SSD-resnet is faster and gives higher quality bounding box than Faster R-CNN.

The implementation is capable of achieving real or near real-time detections, running on a laptop with the characteristics described in Section 3.3. It is also:



(a)



(b)

Figure 6. Experimental results from the implementation of Tensorflow *ssd_resnet*_50_*fpn_coco* model and the *KCF/HOG* tracking algorithm for human in peril in marine environment, (a) during day-time and (b) during night-time.

- Using CPU/GPU, running concurrently with the UAV ground station application.
- Utilizing deep learning techniques for precise distressed human detection.
- Using an advanced object detection technique every N frames of the input video combined with an efficient object tracking algorithm, (tested on both correlation filters and centroid tracking to improve tracking accuracy, in order to evaluate them).
- Applies both a "detection" and "tracking" phase, making it capable of detecting new people and picking up people that may have been "lost" during the tracking phase.

In many human detection methods [32] [33] [34] Depth Similarity Features (DSF) extraction, based on depth information obtained from RGB cameras providing also depth information, is adopted. This information is valuable so as the image depth histograms to be calculated, as it represents the relationship between two local regions. Although there are a lot of differences in the experiments conditions, in used cameras (RGB, RGBD, NIR, FIR), in image resolution used, in human appearance in the scene, in the existence of occlusions or not, in metrics used, etc., an indicative comparison between several methods for human detection and tracking methods is given in Table 3.

Table 2. Implemented algorithms characteristics summarization.

Detection Method	Tracking Method	FPS Throughput Detector & Tracker Cooper.	FPS with MP* Running	Detecting every (N) frames	Mean accuracy
faster_rcnn_inception_v2_coco	Centroid tracker	10	8	1	88.21%
faster_rcnn_resnet50_coco	dlib	19	16	10	90.03%
faster_rcnn_resnet101_v1	KCF/HOG	16	14	10	92.17%
ssd_resnet_50_fpn_coco	KCF/HOG	21	17	10	85.43%

*MP is the Mission Planner software for the UAV control.

Table 3. Indicative comparison between several methods for human detection from rescue UAV using video processing methods.

Research	Human Detection Method	Precision	Accuracy	Throughput	Comments
Our method	faster_rcnn_resnet101_v1 KCF/HOG	95.4	92.17%	16	Combined adaptive detection/tracking
Lestari P. [9]	Viola Jones Algorithm and Chan-Vese active contour detection	99.2	95.3	0.18	-
Xia *et al.* [32]	Depth information combined with 2-D head contour model and a 3-D head surface model	93.5	88.4	0.11	Depth information is taken from a suitable camera
Choi *et al.* [33]	Parameterized heuristics filtering/HOD/SVM	92.5	82.3	0.16	-
Ikemura *et al.* [34]	Relational Depth Similarity Features (RDSF)	90.4	87.5	10	The depth information is obtained from a TOF camera
Chowdhury *et al.* [7]	Background model/ HSI color model and color correlogram/ImHOG	93.5	-	2.17	-
Lee, *et al.* [8]	R-CNN	92.9	-	8	-

(a) (b)

Figure 7. (a) The UAV used during experiments conduction and (b) details of the airborne vision-system.

Last, Figure 7 depicts the UAV used in this research (a) during the take-off phase of a series of experiments and (b) the airborne vision-system, including the vision/NIR camera flying and the gimbal.

6. Conclusions

In this paper, a robust approach for distressed human detection and tracking in several background and illumination conditions has been presented. In particular, the combination of UAVs technologies, GPS/GNSS techniques and advanced image processing algorithms based on CNNs for the instantaneous provision of life-saving services to distressed humans has been integrated and evaluated. The proposed approach includes precise human detection algorithms running every N frame of the input image and an intermediately much faster running human tracking algorithm of OpenCV and various open-source image processing tools. Based on this approach, the fully autonomous rescue UAV is capable of precisely detecting and rescuing distressed humans in numerous terrains or environments under several illumination conditions.

Hardware and software configurations have also been presented in detail. The novelty of this research banks on the generation of a new dataset, suitable for training of the computer vision algorithms, the implementation of advanced image processing algorithms and their seamless integration with a fully autonomous aerial rescue support system. Particularly, in order to acquire a sufficient size dataset, the majority of the training dataset used in this work, are images cropped by videos captured by the author and online available images. Human in peril in numerous terrains and backgrounds under several illumination conditions were filmed by the aim of our own rescue UAV, flying at several altitudes.

Experimental results of the proposed approach are also provided, highlighting its superior performance for the detection and tracking of humans in need of rescue. Evaluation and comparison of the attained accuracy of several utilized

methods and models are also provided.

Conflicts of Interest

The author declares no conflicts of interest regarding the publication of this paper.

References

[1] Seemanthini, K. and Manjunath, S. (2018) Human Detection and Tracking Using HOG for Action Recognition. *Procedia Computer Science*, **132**, 1317-1326. https://doi.org/10.1016/j.procs.2018.05.048

[2] Rakibe, R. and Patil, B. (2013) Background Subtraction Algorithm Based Human Motion Detection. *International Journal of Scientific and Research Publications*, **3**, No. 5. https://doi.org/10.1109/ETCS.2010.440

[3] Darwish, A. Hakami, H. and Chaczko, Z. (2015) Review of Human Motion Detection Based on Background Subtraction Techniques. *International Journal of Computer Applications*, **122**, 1-5. https://doi.org/10.5120/21757-4988

[4] Subudhi, B.N., Ghosh, S. and Ghosh, A. (2013) Change Detection for Moving Object Segmentation with Robust Background Construction under Wronskian Framework. *Machine Vision and Applications*, **24**, 795-809. https://doi.org/10.1007/s00138-012-0475-8

[5] Benezeth, Y., Emile, B., Laurent, H. and Rosenberger, C. (2008) A Real Time Human Detection System Based on Far Infrared Vision. In: Elmoataz, A., Lezoray, O., Nouboud, F. and Mammass, D., Eds., *Image and Signal Processing*, *ICISP* 2008, *Lecture Notes in Computer Science*, Volume 5099, Springer, Heidelberg. https://doi.org/10.1007/978-3-540-69905-7_9

[6] Maier, J. and Humenberger, M. (2018) Movement Detection Based on Dense Optical Flow for Unmanned Aerial Vehicles. *International Journal of Advanced Robotic Systems*, **10**. https://doi.org/10.5772/52764

[7] Chowdhury, S., Kowsar, M. and Deb, K. (2018) Human Detection Utilizing Adaptive Background Mixture Models and Improved Histogram of Oriented Gradients. *ICT Express*, **4**, 216-220. https://doi.org/10.1016/j.icte.2017.11.016

[8] Lee, C., Flynn, M., Vidal, R., Reiter, A. and Hager, G. (2016) Temporal Convolutional Networks for Action Segmentation and Detection. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (*CVPR*), Las Vegas, NV, 26 June-1 July 2016.

[9] Lestari, P. and Schade, H. (2017) RGB-depth Image Based Human Detection Using Viola-Jones and Chan-Vese Active Contour Segmentation. In: *Advances in Signal Processing and Intelligent Recognition Systems*, Springer, Heidelberg, 285-296. https://doi.org/10.1007/978-3-319-67934-1_25

[10] Gonzalez, D. and Hayet, J. (2013) Fast Human Detection in RGB-D Images with Progressive SVM-Classification. In: Klette, R., Rivera, M. and Satoh, S., Eds., *Image and Video Technology*, *PSIVT* 2013, *Lecture Notes in Computer Science*, Volume 8333. Springer, Heidelberg.

[11] Gade, R. and Moeslund, T.B. (2014) Thermal Cameras Applications: A Survey. *Machine Vision and Applications*, **25**, 245-262. https://doi.org/10.1007/s00138-013-0570-5

[12] De Oliveria, D. and Wehrmeister, M. (2018) Using Deep Learning and Low-Cost RGB and Thermal Cameras to Detect Pedestrians in Aerial Images Captured by

Multirotor UAV. *Sensors*, **18**, 2244. https://doi.org/10.3390/s18072244

[13] Lee, J.N. and Kwak, K.C. (2014) A Trends Analysis of Image Processing in Unmanned Aerial Vehicle. *International Scholarly and Scientific Research & Innovation*, **8**, 261-264. https://doi.org/10.5281/zenodo.1090562

[14] Radovic, M., Adarkwa, O. and Wang, Q. (2017) Object Recognition in Aerial Images Using Convolutional Neural Networks. *Journal of Imaging*, **3**, 21-30. https://doi.org/10.3390/jimaging3020021

[15] Lygouras, E., Gasteratos, A., Tarchanidis, K. and Mitropoulos, A. (2018) ROLFER: A Fully Autonomous Aerial Rescue Support System. *Microprocessors and Microsystems*, **61**, 32-42. https://doi.org/10.1016/j.micpro.2018.05.014

[16] Kostavelis, I. and Gasteratos, A. (2017) Robots in Crisis Management: A Survey. 4*th International Conference on Information Systems for Crisis Response and Management in Mediterranean Countries* (*ISCRAM-Med*), Xanthi, Greece, 18-20 October 2017, 43-56. https://doi.org/10.1007/978-3-319-67633-3_4

[17] Quan, A., Herrmann, C. and Soliman, H. (2019) Project Vulture, a Prototype for Using Drones in Search and Rescue Operations. 15*th International Conference on Distributed Computing in Sensor Systems* (*DCOSS*), Santorini, Greece, 29-31 May 2019. https://doi.org/10.1109/DCOSS.2019.00113

[18] Andriluka, M., *et al.* (2010) Vision Based Victim Detection from Unmanned Aerial Vehicles. *Proceedings of the* 2010 *IEEE/RSJ International Conference on Intelligent Robots and Systems* (*IROS*), Taipei, 18-22 October 2010, 1740-1747. https://doi.org/10.1109/IROS.2010.5649223

[19] Blondel, P., Potelle, A., Pégard, C. and Lozano, R. (2014) Fast and Viewpoint Robust Human Detection for SAR Operations. 2014 *IEEE International Symposium on Safety, Security, and Rescue Robotics*, Hokkaido, Japan, 27-30 October 2014. https://doi.org/10.1109/SSRR.2014.7017675

[20] Jingxuan, S., Boyang, L., Yifan, J. and Chih-Yung, W. (2016) A Camera-Based Target Detection and Positioning UAV System for SAR Purposes. *Sensors*, **16**, 1778. https://doi.org/10.3390/s16111778

[21] Bu, F., and Gharajeh, M. (2019) Intelligent and Vision-Based Fire Detection Systems: A Survey. *Image and Vision Computing*, **91**, Article ID: 103803. https://doi.org/10.1016/j.imavis.2019.08.007

[22] Bejiga, M., Zeggada, A., Nouffidj, A. and Melgani, F. (2017) A Convolutional Neural Network Approach for Assisting Avalanche Search and Rescue Operations with UAVs Imagery. *Remote Sensors*, **9**, 100. https://doi.org/10.3390/rs9020100

[23] Pham, H., La, H., Seifer, D. and Nguyen, L. (2018) Reinforcement Learning for UAVs Autonomous Navigation Using Function Approximation. *Proceedings of the* 16*th IEEE International Symposium on Safety, Security and Rescue Robotics* (*SSRR*), Philadelphia, PA, 6-8 August 2018. https://doi.org/10.1109/SSRR.2018.8468611

[24] Niroui, F., Zhang, K., Kashino, Z. and Nejat, G. (2019) Deep Reinforcement Learning Robot for Search and Rescue Operations: Exploration in Unknown Cluttered Environments. *IEEE Robotics and Automation Letters*, **4**, 610-617. https://doi.org/10.1109/LRA.2019.2891991

[25] Sampredo, C., Rodriguez-Ramos, A., Bavle, H., Carrio, A., de la Puente, P. and Campoy, P. (2018) A Fully Autonomous Aerial Robot for Search and Rescue Applications in Indoor Environments Using Learning-based Techniques. *Journal of Intelligent Robotic Systems*, **95**, 601-627. https://doi.org/10.1007/s10846-018-0898-1

[26] Adawadkar, K. (2017) Python Programming-Applications and Future. *International*

Journal of Advanced Engineering and Research Development.
http://ijaerd.com/papers/special_papers/IT032.pdf

[27] Culjak, I., Abram, D., Pribanic, T., Dzapo, H. and Cifrek, M. (2012) A Brief Introduction to OpenCV. 2012 *Proceedings of the* 35*th International Convention MIPRO*, Opatija, Croatia, 21-25 May 2012.

[28] Abadi, M., *et al*. (2016) TensorFlow: A System for Large-Scale Machine Learning. *OSDI* 2016, Savannah, GA, 2-4 November 2016, 265-283.

[29] King, D. (2009) Dlib-ml: A Machine Learning Toolkit. *Journal of Machine Learning Research*, **10**, 1755-1758.

[30] Lin, T., *et al*. (2014) Microsoft COCO: Common Objects in Context. arXiv 2014, arXiv:1405.0312. https://doi.org/10.1007/978-3-319-10602-1_48

[31] Huang, J., *et al*. (2012) Speed/Accuracy Trade-Offs for Modern Convolutional Object Detectors. arXiv 2017, arXiv:1611.10012.
https://doi.org/10.1109/CVPR.2017.351

[32] Xia, L., Chen, C. and Aggarwal, J.K. (2011) Human Detection Using Depth Information by Kinect. *CVPR* 2011 *Workshops*, Springs, CO, 20-25 June 2011, 15-22.
https://doi.org/10.1109/CVPRW.2011.5981811

[33] Choi, B., Mericli, C., Biswas, J. and Veloso, M. (2013) Fast Human Detection for Indoor Mobile Robots Using Depth Images. 2013 *IEEE International Conference on Robotics and Automation* (*ICRA*), Karlsruhe, Germany, 6-10 May 2013.
https://doi.org/10.1109/ICRA.2013.6630711

[34] Ikemura, S. and Fujiyoshi, H. (2010) Real-Time Human Detection Using Relational Depth Similarity Features. In: Kimmel, R., Klette, R. and Sugimoto, A., Eds., *Computer Vision-ACCV* 2010. *Lecture Notes in Computer Science*, Volume 6495, Springer, Heidelberg. https://doi.org/10.1007/978-3-642-19282-1_3