Towards an Machine Learning-Based Edge Computing Oriented Monitoring System for the Desert Border Surveillance Use Case

KHALIFA M. BELLAZI1, RODRIGO MARINO2, (Graduate Student Member, IEEE), JOSE M. LANZA-GUTIERREZ2, AND TERESA RIESGO1, (Member, IEEE)

1Centro de Electrónica Industrial, Escuela Técnica Superior de Ingenieros Industriales, Universidad Politécnica de Madrid, 28040 Madrid, Spain
2Department of Computer Science, University of Alcalá, 28871 Alcalá de Henares, Spain

Corresponding author: Khalifa M. Bellazi (k.bellazi@alumnos.upm.es)

This work was supported in part by the Libyan Ministry of Education, and in part by the Universidad Politécnica de Madrid through a predoctoral contract (Programa propio) by Banco Santander S.A. under Grant RR23/2017.

ABSTRACT The design of border surveillance systems is critical for most countries in the world, having each border specific needs. This paper focuses on an Internet of Things oriented surveillance system to be deployed in the Sahara Desert, which is composed of many unattended fixed platforms, where the nodes in the edge have a Forward Looking InfraRed (FLIR) camera for field monitoring. To reduce communications and decentralise the processing, IR images should be fully computed on the edge by an Automated Target Recognition (ATR) algorithm, tracking and identifying targets of interest. As edge nodes are constrained in energy and computing capacity, this work proposes two ATR systems to be executed in low-power microprocessors. Both proposals are based on using Bag-of-Features for feature extraction and a supervised algorithm for classification, both differing in segmenting the InfraRed image in regions of interest or working directly with the whole image. Both proposals are successfully applied to infer about a dataset generated to this end, getting a trade-off between computing cost and detection capacity. As a result, the authors obtained a detection capacity of up to 97% and a frame rate of up to 5.71 and 59.17, running locally on the edge device and the workstation, respectively.

INDEX TERMS Automatic target recognition, approximate computing, bag of features, border surveillance, classification, edge computing, Internet of Things, machine learning, Sahara desert.

I. INTRODUCTION

State borders are regions where multiple flows of resources and people converge, for instance, traffic or electrical wiring. Their wide extension increases the complexity of monitoring the activities performed in the border. Some issues of concern are related to the type of flows and their identification, such as traffic monitoring, abnormal events in border installations, or illegal activities. Thus, governments make continuous efforts to invest in accurate border monitoring systems to mitigate these risks. For instance, abnormal events in pipeline gas installations, such as gas leakage, can be tracked and then avoid the associated environmental pollution and economic costs [1]. One drawback of real-world borders is that they offer a plethora of scenarios with different needs:

- volume and type of traffic, orography, area to cover, weather conditions, as well as available economic resources to deploy the system [2]. Hence, the use case requirements play a major role in the design of these systems. One scenario that is not widely covered in the literature is the desert borders whose environmental conditions make not suitable for traditional monitoring approaches. Hence, this work focuses on a border traffic monitoring use case based on the Sahara Desert conditions, whose border has a wide-area with a low traffic volume [3].

Considering the broad extension of the Libia Sahara border, Internet of Things (IoT) appears as an infrastructure to increase the areas covered, while reducing the human intervention within the monitoring operation. IoT incorporates ubiquitous computing which deploys sensors, processing systems, and communications for detecting targets in the border [4]. Therefore, this information might be shared or...
analysed by different nodes from multiple layers. Artificial Intelligence (AI) techniques might process this information to create autonomous decision-making algorithms in the IoT infrastructure. This scenario would enhance the response time, and also the needs of trained personnel [5].

Traffic monitoring sensing systems are based on cameras, generating a high amount of sensitive data. Therefore, transmitting data through the network would increase energy consumption and would produce privacy problems [6]–[8]. To alleviate the network backbone and avoid privacy losses, a solution is to compute the target detection in the IoT edge layer. To this end, the use case in this paper considers a hypothetical monitoring system composed of many unattended fixed surveillance platforms with long-range sensing capabilities and the hardware needed to manage, process, and send the data through long-range wireless communication. Each surveillance platform is intended to be designed within an Internet of Things (IoT) focus [9], where the nodes in the edge have a Forward Looking InfraRed (FLIR) camera. This sensing technology is useful in detecting heat a few miles away and is especially powerful when the area has a low density of vegetation as occurs in the Sahara Desert [10].

Automated Target Recognition (ATR) algorithms are decision engines which track and identify targets of interest within the field of view. In the context of IoT, decision engines are usually placed in the cloud layer, as they require high resource and consumption capabilities [4]. According to the use case, privacy leakage might be avoided by moving the decision engines to the edge layer [11]. Therefore, the nodes in the edge will have to fully compute ATR algorithms to track and identify targets of interest within the field of view.

The inclusion of powerful machine learning techniques inside the ATR decision engines has gained more relevance due to their performance capabilities compared to traditional ATR with limited AI. However, this type of techniques has high computational cost [12]. Hence, this paper tries to delve into the open problem of moving the machine learning-based ATR systems from the cloud to the edge domain to robust the system, while maintaining privacy and machine learning performance. As the nodes in the edge layer are constraint in energy and computing capacity, decision engine algorithms should be carefully studied and adapted if needed to reduce the energy cost, while maintaining valid response times for the application [6]. Under this focus, much effort must be put into detecting and classifying moving targets, ignoring stationary ones to save energy and computing needs, lowering deployment and maintenance costs. This fact, along with a low occurrence of moving targets, means that the monitoring system could be designed considering a low processing speed and a low need for transmitting data over the network. Thus, instead of sending all the sensed data, the system could only send alerts, also avoiding privacy problems.

On this basis, this paper proposes two edge-oriented machine learning-based ATR systems which can provide a low-power monitoring solution using IR images to be incorporated in the edge nodes. The two ATR systems proposed are executed in a low-power microprocessor whose evaluation is based on the border surveillance Sahara desert use case, even though it could be extended to other monitoring use cases, such as pipeline leakage detection [1]. The fact of using a low-power microprocessor in this environment with extreme temperatures is especially interesting because of the low-energy dissipation, meaning that the system will be more reliable due to the absence of refrigeration means. Due to the InfraRed (IR), images provided by FLIR cameras have a lower resolution and quality than images from the visible field, the authors propose to take special attention to the feature extraction stage by considering a method derived from the text mining field, the Bag-of-Features (BoF) [13], [14] method, in combination with usual feature descriptors. Thus, once features are extracted according to the BoF strategy, the content in the IR image is classified based on a supervised algorithm. The two ATR systems proposed use IR image segmentation based on Regions Of Interest (ROIs). They mainly differ whether working directly with the whole image or not, which could mean a lower computing cost but a possible loss in performance, fitting with the approximate computing focus in edge computing.

The system validation of the proposal, first, is based on the detection capacity by inferring about a dataset, specially generated for the use case. To this end, we focus on some usual performance metrics in machine learning, such as f-score, recall, and precision [15]. During this analysis, the authors consider three usual feature descriptors to generate the BoF model: Speeded Up Robust Features (SURF), Scale Invariant Feature Transform (SIFT), and Oriented fast and Rotated brief (ORB) [16]–[18]; and four supervised algorithms: Support Vector Machine (SVM), K-Nearest-Neighbor (KNN), Naive Bayes (NB), and Decision Tree (DT) [15]. The combination of different BoF descriptors and supervised algorithms provides a space of solutions for ATR in IR images. Second, the two ATR systems are executed in an edge node getting a Pareto trade-off between computing cost and detection capacity to facilitate the decision making under an approximate multi-objective computing focus [19]. As a result, the authors obtained both detection capacity metrics and computing times within the state-of-the-art as will be discussed. The main contributions of this work are summarised as follows:

- A proposal of two edge-oriented machine learning-based ATR approaches to support low-power ATR based on IR images.
- An exploration of solutions from the proposed systems to provide an ATR system suitable to the desert border monitoring use case.
- The implementation of the solutions inside a resource-constrained and low-power edge device demanded in border monitoring applications.

The remainder of this work is structured as follows. Section II discusses the background. The two edge-oriented machine learning-based ATR systems proposed are described in Section III. The experimental methodology and the experimental results are exposed in Sections IV and V, including a
comparison with the state-of-the-art. Conclusions are left for Section VI.

II. BACKGROUND IN SURVEILLANCE SYSTEMS
This section introduces some background in surveillance systems, including discussion on mobile and fixed surveillance platforms, sensing sources, and related works in ATR.

A. MOBILE AND FIXED SURVEILLANCE PLATFORMS
With the recent development of unmanned aerial vehicles as drones, border surveillance systems can be installed and operated in mobile devices which include specialized sensors and advanced navigation systems. However, this type of mobile systems is not adequate for long borders, as occurs in the Sahara Desert use case. Most limitations are related to the range and the impossibility of guaranteeing border protection at all times [20]–[23].

Fixed surveillance platforms could play an interesting role as an alternative to mobile systems. However, this solution is restricted because fixed towers, needed to be deployed along the border, have a high cost. The deployment of this technology efficiently is relevant to reduce costs, as some authors asserted [24], [25]. In our use case, this limitation is addressed by i) considering an edge-oriented machine learning-based ATR system to be executed in a low-power device in the tower, reducing costs in networking and computing capacity, and ii) using long-range sensors, reducing the number of towers needed along the border.

B. SENSING SOURCES
Two main technologies are usually considered as sensing sources in border surveillance: Synthetic Aperture Radar (SAR) and FLIR cameras. SAR is a type of radar used to create two-dimensional or three-dimensional reconstructions of objects and is typically mounted on moving platforms [26]. FLIR cameras detect IR radiation to get an image of a target area and are mounted on both fixed and moving platforms, having two main types: Medium-Wave Infrared (MWIR) and Long-Wave Infrared (LWIR) cameras, where LWIR cameras are useful for detecting hot engine parts and human body heat a few miles away [27].

In the context of our use case, the sensing technology used is based on LWIR cameras placed on the top of the towers, getting IR images about miles away in the vicinity. This focus is related to the needed of reducing the number of towers deployed, with an estimate of one tower every four kilometers. Other sensing technologies could be incorporated into the use case for implementing physical barriers in the terrain as a redundant warming detection system [28], [29].

C. AUTOMATED TARGET RECOGNITION BASED ON INFRARED IMAGERY
The process of detecting and classifying targets based on sensing sources is one of the most important challenges in border surveillance systems. This identification task could be done manually by expert humans. However, the use of complex sensing sources as the ones introduced before, the huge amount of data generated, especially for long borders, and response time constraints imply that this identification task must be performed automatically.

The purpose of ATR is that given any input (e.g., an LWIR image) the system should be able to automatically match the content in the input with a predefined set of classes using an intelligent decision engine. To this end, supervised classification methods, from the machine learning field, are usually considered. Thus, an inference function is built by examples, being later used for mapping future samples. As it is well-known, the success of CBIR (Content-Based Image Retrieval) systems is much influenced by the selection of the learning algorithm, as well as the way in which feature extraction is performed [30].

The design of ATR systems for the desert boundary has not been widely studied in the literature. As far as the authors know, there is only work with a focus similar to ours was proposed by Briones et al. [31]. In this work, they proposed a radar system in combination with a decision engine based on an Artificial Neural Network (ANN) to classify several targets, which were pedestrians, trucks, and aeroplanes, getting a detection capacity of 80%. The work in [31] includes some limitations for a low-power approach as the one considered in this paper, these are i) ANN are usually executed in many-core oriented processors, such as Graphics Processing Units (GPUs), which are power demanding devices and ii) a radar system consumes twice the energy that an IR camera. Moreover, radar systems are more prone to failures due to they have moving parts. Therefore, this approach is not directly applicable in the edge layer for the desert use case.

Outside the desert use case, there are some interesting state-of-the-art works in the ATR field based on IR imagery, which are summarized in Table 1 and described below:

- Sun et al. [32] (2014) proposed a Concave-Convex Partition (CCP) strategy to improve Local Binary Pattern (LBP) and Local Ternary Pattern (LTP) features by dividing both into two groups, i.e., concave and convex, according to the contrast between local and global intensities. As a result, the authors obtained a detection capacity from 65% to 91% with a frame rate from 41 to 15, respectively, for ten classes of military vehicles, running on a 3.41 GHz AMD Phenom II processor.

- Khan et al. [33] (2014) proposed to apply Bag-of-Words (BoW) in conjunction with Histogram of Oriented Gradients (HOG) for feature extraction. As a result, the authors concluded that BoW-based recognition is efficient and robust in the context of the use case with ten classes for military vehicles, getting an accuracy higher than 99%. The authors also identified that the detection capacity of the proposal is affected by the size of the vocabulary generated, where the detection capacity in the worst case was 85%.

- Christiansen et al. [34] (2014) contributed to the ATR for thermal imaging to detect animals in agricultural mowing operations. To this end, the authors proposed...
a novel thermal feature extraction algorithm, where the thermal signature was calculated using morphological operations for feeding a KNN classifier in a binary domain. As a result, the authors obtained an accuracy of 84.7% in an altitude range of 3–10 m and an accuracy of 75.2% for an altitude range of 10–20 m.

- Yu et al. [35] (2015) proposed a technique for joint recognition, segmentation and pose estimation. The problem was formulated in a probabilistic level set framework where a shape constrained generative model was used to provide a multi-class and multi-view shape prior and where the shape model involved a Couplet of View and Identity Manifolds (CVIM). For realistic target chips, the authors adopted a Gradient-Boosted Particle Swarm Optimization (GB-PSO), resulting in a detection capacity of up to 78% with a dataset of eight classes for military vehicles and a frame rate of 0.16 running on a 2.5GHz Intel Quad-Core processor.

- Rodger et al. [36] (2016) trained a Convolutional Neural Network (CNN) to demonstrate enhanced target recognition and improved false alarm rates in IR imaging. As a result, the authors achieved an accuracy higher than 95% for six classes related to land defence, including aerial and land vehicles, as well as people.

- Rujikietgumjorn and Watcharapinchai [37] (2017) proposed a pedestrian detection method for IR images, performing in real-time on an edge device. To this end, the authors considered HOG for feature extraction in ROIs and SVM for classification. As a result, they achieved a detection capacity of 60% in the binary problem, getting a frame rate of 12.2 on a Raspberry Pi 3b device.

- Millikan et al. [38] (2018) introduced two methods integrating the discrimination capability of Quadratic Correlation Filters (QCF) into a CNN. As a result, the authors checked that the proposal not only improved the detection capacity, getting an accuracy of 98% for ten classes of military vehicles, but also the proposal assisted in target localization.

- Kim et al. [39] (2018) presented an ATR fusion strategy based on SAR and IR sensing data. To this end, the authors proposed a doubly weighted neural network fusion scheme at the decision level. As a result, the authors obtained an accuracy higher than 95% for sixteen classes related to military and non-military vehicles, getting a frame rate of 200 running on a GPU controlled by an Intel Core i7-5820 processor with 128 GB RAM.

- Sheeny et al. [40] (2018) proposed four methods to detect non-military vehicles in polarized LWIR imagery under a binary problem. The four methods were based on neural networks: MobileNet SSD, InceptionV2 SSD, Faster R-CNN Resnet-50, and Faster R-CNN Resnet-101. The authors implemented all the methods in an NVIDIA Titan X GPU controlled by an Intel Core i7-7700HQ processor with 32GB RAM. As a result, they concluded that the most accurate method was Faster R-CNN Resnet-101 with 80.94%, but with a frame rate of 6.6. On the contrary, the quickest method was MobileNet SSD with a frame rate of 53.4, but with an accuracy of 58.56%.

- Gomez et al. [41] (2018) developed a people counting algorithm on IR images based on CNNs small enough that they ran on a limited-memory low-power platform. As a result, the authors achieved a detection capacity of 96%, for the binary problem, with a frame rate of 0.007 running on an ARM Cortex M4 processor.

- Nasrabadi [42] (2019) proposed a double-neural network approach for ATR of ten classes of military vehicles. First, the authors computed the image provided by an IR camera by a fully CNN to extract potential targets. Then, the potential target points were introduced into a deep CNN classifying them into different types. The proposal provided an accuracy of 98% and a frame rate of 11, running on an NVIDIA Titan X GPU.

The main differences between our proposal and the works outside the desert focus for ATR systems based on IR imagery in Table 1 are as follows:

- Most of the works focused on identifying vehicles [32], [33], [35], [38]–[40], [42], others focused on detecting the presence of animals [34] or people [37], [41] by following a binary classification approach, and only one of the works considered vehicles and people in the same classification problem [36]. Our approach focuses on detecting targets within the three classes in the same system (animals, people, and vehicles).
because these are the type of targets crossing the border.

- Most of the proposals were evaluated using a regular processor [32], [35] or a powerful GPU [39], [40], [42], providing frame rates which could limit the real-time applicability of the algorithm in a low-power edge device. Other proposals did not discuss the computing platform used nor any metric for analyzing the computing cost, which limits the comparison [33], [34], [36], [38]. Moreover, some of these works [36], [38] applied ANN without any optimization for low-power devices, which means that both works are related to powerful GPUs. Thus, only the works in [37], [41] designed algorithms ready to be executed in low-power devices where [37] and [41] obtained a frame rate of 0.007 and 12.6 in a binary classification problem, respectively. Note that the work in [41] considered a CNN, being specially adapted to work in the constrained device, but having a low frame rate. In our proposal, the authors focus on designing a multi-class ATR system ready to be executed on a constrained edge device with an adequate frame rate for the application.

Outside ATR systems based on IR imagery, the state-of-the-art is full of successful works applying feature extraction and classification methods for images in the visible field, as in [43], [44]. Most of these works consider deep learning approaches, providing an excellent detection capacity but needing to be executed in powerful GPUs, with frame rates which limit the applicability in the edge. For instance, Zhao et al. [45] recently analyzed eighteen deep learning approaches with excellent results, but having frame rates from 0.03 to 46, running under an NVIDIA Tesla controlled by an Intel i7 processor. The authors also find approaches specially designed to work within the edge layer. For instance, Nikouei et al. [46] proposed a lightweight CNN for pedestrian detection, being executed by an ARMv7 1.2GHz processor, having an average false positive rate of 6.6% and a frame rate of 1.82. Nikouei et al. [47] improved the work in [46] by including a decision tree based hybrid kernelized correlation filter algorithm, getting a higher detection capacity and a similar frame rate as before. Other interesting works in this line can be found in the recent survey in [48] for human detection at the edge, having frame rates from 0.4 to 14.

On this basis, this paper is within the edge computing and approximate computing fields of knowledge, which aroused in the last years [6], [49] and whose objective is to bring the computation closer to the sensing place. The novelty in this paper is the possibility of embedding directly in the edge the capacity of detecting targets of interest in a specific border surveillance use case based on LWIR imagery, using to this end constrained hardware in computing capacity and with the purpose of getting both detection capacity and frame rate metrics within the state-of-the-art. As a way of emphasizing the applicability in the edge, the two proposals are executed in an edge device to get a trade-off between computing cost and detection capacity to facilitate the decision making under an approximate computing focus through the Pareto theory.

As far as the authors know, this is the first study within the border surveillance field and IR sensing, which includes concepts from the edge computing and approximate computing fields.

III. MACHINE LEARNING-BASED ATR SYSTEMS PROPOSED

This section describes the two machine learning-based ATR systems proposed in this paper, i.e., the bounding box prediction and the frame-based prediction methods. To this end, the authors first describe several aspects in common for both approaches: the tasks of feature extraction and object classification.

A. FEATURE EXTRACTION AND OBJECT CLASSIFICATION

For the two ATR systems proposed, once a movement is detected in the IR video sequence, the next step is to extract features of interest from the binarized image with the moving target using the BoF approach. Then a classifier uses the BoF result to identify the class of the object inside the image.

In the context of computer vision, BoF represents an image through a histogram of visual words. However, BoF cannot work directly with images, being necessary to apply a transformation as follows: each image is abstracted by several characteristic local patches, which extract the information of interest provided by a feature descriptor. In this use case, we test three different feature descriptors (SIFT, SURF, and ORB) intending to establish a trade-off between their computing cost and its effect in the detection capacity. For a specific descriptor, the information in the patch is encoded in a vector form (a word in the context of BoF) with size determined by the descriptor itself.

BoF requires generating a dictionary in a previous training step. To this end, a set of words are generated as discussed before for a group of images. The dictionary is created through an unsupervised machine learning algorithm, usually, k-means, analysing the set of words. The size of this dictionary is given by the number of clusters (codewords) identified by the unsupervised algorithm. Each word of an image is associated with a codeword of the dictionary. As a result, the image is represented by a histogram of codewords.

Once the BoF model is created, it is possible to express a future image in a histogram of codewords as follows: i) the image is abstracted by several words. ii) each word is associated with a codeword. iii) the histogram is generated based on the codewords selected.

Finally, a supervised machine learning approach is used for identifying the target from the image according to the histogram generated. As expected, this approach requires two steps: training and inference. The training stage sets the parameters of a machine learning model for making a prediction through a histogram. In this regard, a collection of codewords histograms from different images and their target labels are required. The next step, inference, consists in making a prediction using the histogram from an image.
B. BOUNDING BOX PREDICTION

This section includes the first machine learning-based ATR system proposed, which is shown in Fig. 1. The system is characterized by segmenting the input IR video stream into ROIs, also named bounding boxes, where possible targets are. As usual, the machine learning-based ATR system is divided into two stages: training, where the parameters of the BoF and the supervised learning algorithm are adjusted, and prediction, where the class of the input ROI is inferred.

The training stage is designed to be executed offline at a workstation. The procedure starts by collecting a set of labelled binary images (masks, see Fig. 2) corresponding to examples of ROIs. These images are considered to train the BoF feature extraction method as follows:

1) Each image in the collection is abstracted by several characteristic local patches, based on a feature descriptor, and encoded in form of a word.
2) All words are used to create a dictionary through k-means, obtaining the BoF model.
3) Each image represented by a histogram of codewords. The histogram of codewords and the label for each image in the set are used to train the supervised classification model.

The prediction stage is designed to be executed online on the edge device. The procedure starts by detecting moving frames in the input IR video stream so that only frames of interest are analysed, saving computing capacity and energy. To this end, the authors consider the improved three-frame difference model exposed in [50], [51], characterized by a...
K. M. Bellazi et al.: Toward an Machine Learning-Based Edge Computing Oriented Monitoring System

**FIGURE 2. Examples for the datasets considered.**

(a) Example for dt_13_eq_ind.

(b) Example for dt_3_eq_ind.

(c) Example for dt_3_diff_ind.

(d) Example for dt_6_eq_gr.

Reduced computational cost and noise. Note that image binarization is performed in this point as described in the three-frame difference model. Once a moving frame is detected, ROIs are extracted by getting contours in the current frame, where each ROI is the smallest rectangle containing a contour. Then, a BoF histogram is generated for each ROI according to the previously obtained codewords. Next, the classification algorithm infers the class of the possible target in the ROI, which is the output of the system.

**C. FRAME-BASED PREDICTION**

This section includes the second machine learning-based ATR system proposed, which is shown in Fig. 1(b). The system is characterized by not segmenting the IR video stream into ROIs, using instead the full frames. As before, the system is divided into two stages: training and prediction.

The training stage is designed to be executed offline on a workstation. The procedure starts by collecting a set of labelled binary images corresponding to examples of full frames. In case that there is more than one type of target in a full frame, the label will refer to the most dominant content to the detriment of others. These images are considered to train the BoF model. Next, the histogram of codewords and the label for each image in the collection are considered to train the supervised classification algorithm.

The prediction stage is designed to be executed online in the edge device. The procedure starts by detecting moving frames in the input IR video stream, according to the three-frame difference model. Once a moving frame is detected, a BoF histogram is generated for the full frame without segmentation, based on the obtained codewords. Next, the classification algorithm infers the class of the
possible target in the full frame, which is the output of the system.

**IV. EXPERIMENTAL METHODOLOGY**
This section discusses the detection capacity metrics and the datasets considered to evaluate the two systems proposed.

**A. DETECTION CAPACITY METRICS**
The authors consider some usual metrics in machine learning when comparing the performance of classification methods. These metrics are:

- **Recall** (also called true positive rate or sensitivity), which is calculated as
  \[
  \text{recall} = \frac{TP}{TP + FN},
  \]
  where TP and FN are the numbers of true positives and false negatives. The higher the recall, the algorithm performs better in avoiding false negative cases.
- **Precision** (also called positive predictive value), which is calculated as
  \[
  \text{precision} = \frac{TP}{TP + FP},
  \]
  where FP is the number of false positives. The higher the precision, the algorithm performs better in avoiding false positive cases.
- **F-score**, which combines precision and recall metrics, is calculated as
  \[
  \text{f-score} = \frac{2 \cdot \text{recall} \cdot \text{precision}}{\text{recall} + \text{precision}}.
  \]
  The higher the f-score, the trade-off between precision and recall is maximized.

The authors also consider the confusion matrix visual method to represent the performance of the classifiers. A confusion matrix is a square matrix whose number of columns and rows equals to the number of classes in the classification problem. Each row shows the number of instances for a particular predicted class, and each column indicates the number of instances for a particular true class from that specific row. Thus, each cell contains the rate of matches between the predicted class and the true class. An ideal confusion matrix has a value of 1.0 in the diagonal, while the remaining positions take a value of 0.0. In the diagonal In such a case, all instances are classified correctly according to the true class label provided.

**B. DATASET DESCRIPTION**
There are many datasets in the literature for object recognition, such as MPEG-7 [52], MSRA10K [53], and MCL [54]. However, the authors did not find any freely available dataset fitting our use case characterized by i) the usage of low-resolution silhouette IR images (masks) and ii) the identification of up to thirteen useful classes, grouped by three main categories: animals, people, and vehicles. Thus, for the system validation, four datasets oriented to Sahara desert Libya border have been generated focusing on targets that can potentially cross is based on the two previous considerations.

The four datasets, which are exemplified in Fig. 2, were generated based on collecting images from MPEG-7, MSRA10K, and MCL datasets for the classes matching our use case. The images for the remaining classes were generated through an LWIR video stream, where frames of interest were extracted using the improved three-frame difference algorithm discussed before. In this video stream, only frames containing targets of interest were labelled and added to our dataset. The four datasets are as follows:

- **dt_13_eq_ind** includes 624 binary images from thirteen categories of individual targets (army-man, camel, car, cat, cow, deer, dog, elephant, giraffe, horse, person, walking-person, and truck). Images have the same size of 300 \( \times \) 300 pixels and targets in the same class have a similar size.
- **dt_3_eq_ind** includes 750 binary images from three categories of individual targets (people, vehicles, and animals). Images have the same size of 480 \( \times \) 270 pixels and targets in the same class have a similar size.
- **dt_3_diff_ind** includes 810 binary images from three categories (people, vehicles, and animals). Images have the same size of 480 \( \times \) 270 pixels and targets in the same class can have a significantly different size.
- **dt_6_eq_gr** includes 1500 binary images from six categories combining individual and group targets (animal, vehicle, person, a group of animals, a group of vehicles, and a group of people). Images have the same size of 480 \( \times \) 270 pixels and targets in the same class have a similar size.

**V. EXPERIMENTAL RESULTS**
This section first evaluates the two systems proposed in detection capacity. To this end, the authors encoded the two systems in Python 2.7, considering both OpenCV 3.0 and Scikit-learn libraries, running the algorithms in a regular workstation. Next, the two systems are evaluated in an edge device, comparing the results obtained to the state-of-the-art.

**A. EVALUATING THE BOUNDING BOX PREDICTION METHOD IN DETECTION CAPACITY**
This section analyzes the performance of the bounding box prediction method while using SURF, SIFT, and ORB descriptors applied to four different classification algorithms: SVM, KNN, DT, and NB.

To this end, the authors consider the three databases with individual targets described before, i.e., dt_13_eq_ind, dt_3_eq_ind, and dt_3_diff_ind. The reason for only considering the databases with individual targets is because the authors are interested in analyzing the performance of the algorithm in a single ROI. Note that the bounding box method includes a specific procedure for extracting and classifying each ROI independently. Besides, for each of the three databases, the authors considered a holdout strategy to randomly select the 80% of samples for training,
TABLE 2. Performance metrics applying the bounding box prediction method to dt_13_eq_ind.

| Target          | SVM BoF with SIFT | SVM BoF with ORB | SVM BoF with SURF | KNN BoF with SIFT | KNN BoF with ORB | KNN BoF with SURF | Naive Bayes BoF with SIFT | Naive Bayes BoF with ORB | Naive Bayes BoF with SURF | DT BoF with SIFT | DT BoF with ORB | DT BoF with SURF |
|-----------------|-------------------|------------------|-------------------|-------------------|------------------|------------------|------------------------|------------------------|------------------------|----------------|----------------|----------------|
|                 | precision | recall | f-score | precision | recall | f-score | precision | recall | f-score | precision | recall | f-score | precision | recall | f-score | precision | recall | f-score | precision | recall | f-score | precision | recall | f-score |
| Army man        | 0.73     | 0.80   | 0.76   | 0.75     | 0.90   | 0.82   | 0.88     | 0.70   | 0.78   | 10.00      | 0.70     | 0.78   | 10.00      | 0.70     | 0.78   | 10.00      | 0.70     | 0.78   | 10.00      | 0.70     | 0.78   | 10.00      |
| Camel           | 0.25     | 0.20   | 0.22   | 0.45     | 0.30   | 0.35   | 0.67     | 0.40   | 0.50   | 10.00      | 0.70     | 0.78   | 10.00      | 0.70     | 0.78   | 10.00      | 0.70     | 0.78   | 10.00      | 0.70     | 0.78   | 10.00      |
| Car             | 1.00     | 0.90   | 0.90   | 1.00     | 0.90   | 0.90   | 1.00     | 0.90   | 0.90   | 10.00      | 0.90     | 0.90   | 10.00      | 0.90     | 0.90   | 10.00      | 0.90     | 0.90   | 10.00      | 0.90     | 0.90   | 10.00      |
| Cat             | 0.90     | 0.90   | 0.90   | 0.67     | 0.80   | 0.73   | 0.89     | 0.80   | 0.84   | 10.00      | 0.80     | 0.84   | 10.00      | 0.80     | 0.84   | 10.00      | 0.80     | 0.84   | 10.00      | 0.80     | 0.84   | 10.00      |
| Cow             | 0.83     | 1.00   | 0.91   | 1.00     | 0.90   | 0.95   | 0.77     | 1.00   | 0.88   | 10.00      | 1.00     | 0.88   | 10.00      | 1.00     | 0.88   | 10.00      | 1.00     | 0.88   | 10.00      | 1.00     | 0.88   | 10.00      |
| Deer            | 0.77     | 1.00   | 0.90   | 0.91     | 1.00   | 0.95   | 0.83     | 1.00   | 0.99   | 10.00      | 1.00     | 0.99   | 10.00      | 1.00     | 0.99   | 10.00      | 1.00     | 0.99   | 10.00      | 1.00     | 0.99   | 10.00      |
| Dog             | 0.71     | 0.50   | 0.59   | 0.58     | 0.70   | 0.64   | 0.75     | 0.60   | 0.67   | 10.00      | 0.60     | 0.67   | 10.00      | 0.60     | 0.67   | 10.00      | 0.60     | 0.67   | 10.00      | 0.60     | 0.67   | 10.00      |
| Elephant        | 0.78     | 0.70   | 0.74   | 1.00     | 0.80   | 0.89   | 0.82     | 0.90   | 0.86   | 10.00      | 0.90     | 0.86   | 10.00      | 0.90     | 0.86   | 10.00      | 0.90     | 0.86   | 10.00      | 0.90     | 0.86   | 10.00      |
| Giraffe         | 0.91     | 1.00   | 0.95   | 1.00     | 1.00   | 1.00   | 0.83     | 1.00   | 0.91   | 10.00      | 1.00     | 0.91   | 10.00      | 1.00     | 0.91   | 10.00      | 1.00     | 0.91   | 10.00      | 1.00     | 0.91   | 10.00      |
| Horse           | 0.56     | 0.50   | 0.53   | 0.62     | 0.50   | 0.56   | 1.00     | 0.80   | 0.89   | 10.00      | 1.00     | 0.89   | 10.00      | 1.00     | 0.89   | 10.00      | 1.00     | 0.89   | 10.00      | 1.00     | 0.89   | 10.00      |
| Person          | 0.80     | 0.80   | 0.80   | 0.80     | 0.80   | 0.80   | 0.82     | 0.80   | 0.82   | 10.00      | 0.80     | 0.82   | 10.00      | 0.80     | 0.82   | 10.00      | 0.80     | 0.82   | 10.00      | 0.80     | 0.82   | 10.00      |
| Person walking  | 0.91     | 1.00   | 0.95   | 0.67     | 1.00   | 0.80   | 1.00     | 1.00   | 1.00   | 10.00      | 1.00     | 1.00   | 10.00      | 1.00     | 1.00   | 10.00      | 1.00     | 1.00   | 10.00      | 1.00     | 1.00   | 10.00      |
| Truck           | 0.67     | 0.40   | 0.50   | 0.78     | 0.70   | 0.74   | 0.78     | 0.70   | 0.74   | 10.00      | 0.70     | 0.74   | 10.00      | 0.70     | 0.74   | 10.00      | 0.70     | 0.74   | 10.00      | 0.70     | 0.74   | 10.00      |

Average/Total: 0.73/0.73/0.73/0.78/0.78/0.78/0.82/0.82/0.82/130

Starting with dt_13_eq_ind, Table 2 shows precision, recall, and f-score for each pair descriptor-classification algorithm. Note that the support metric is also shown, referring to the number of true samples in a class. In this table, SVM with SURF provides the best performance with up to 82% for precision, recall, and f-score metrics. The best results for the other algorithms are given by KNN with SIFT, NB with SURF, and DT with SURF, with up to 61%, 67%, and 68% for all metrics, respectively. Thus, SVM outperforms the other algorithms in this first dataset.
Focusing on the algorithm with the best performance, Fig. 3 shows the confusion matrices for this first dataset. In this figure, the authors check that the method fails in many animal cases. For instance, it is especially interesting the behaviour of the system for the camel class because the performance metrics are low in comparison to the other classes. This behaviour is related to the low resolution of LWIR images, meaning that objects with similar appearances cannot be accurately identified. That means that a camel could be confused, for instance, with a cow because both have four legs attached to a body. Thus, specific targets, such as truck, cow, or giraffe, are not suitable when using low-resolution LWIR images, instead, we might group these targets in a more generic class. As a consequence of this conclusion, the targets were grouped into three generic groups: animals, vehicles and people.

Following the focus of using more generic classes, Table 3 shows precision, recall, and f-score for each pair descriptor-classification algorithm, while inferring based on the dt_3_eq_ind database. Analyzing this table, we check that the three metrics are significantly improved for all the cases, where the SURF-based approach with SVM provides the best performance with a value of up to 97% for precision, recall, and f-score. The best results for the other algorithms are given...
by KNN with ORB, NB with ORB, and DT with SURF, with up to 95%, 87%, and 89% for all metrics, respectively. Thus, SVM outperforms the other algorithms in this second dataset. Focusing on the algorithm with the best performance, Fig. 4 shows the confusion matrices for this second dataset. Analyzing this figure, the authors check that confusion between classes was reduced in comparison with Fig. 3, resulting in a more robust system.

Following the focus of using more generic classes, Table 3 shows precision, recall, and f-score for each pair descriptor-classification algorithm, while inferring based on the dt_3_eq_ind database. Analyzing this table, we check that
TABLE 4. Performance metrics applying the bounding box prediction method to dt_3_diff_ind.

| Target | SVM (BoF with SIFT) | SVM (BoF with ORB) | SVM (BoF with SURF) | KNN (BoF with SIFT) | KNN (BoF with ORB) | KNN (BoF with SURF) | NB (BoF with SIFT) | NB (BoF with ORB) | NB (BoF with SURF) | DT (BoF with SIFT) | DT (BoF with ORB) | DT (BoF with SURF) |
|--------|--------------------|--------------------|--------------------|--------------------|--------------------|--------------------|--------------------|--------------------|--------------------|--------------------|--------------------|--------------------|--------------------|
|        | precision | recall | f-score | precision | recall | f-score | precision | recall | f-score | precision | recall | f-score | precision | recall | f-score | support |
| Animal | 0.95     | 0.98   | 0.97   | 0.98     | 1.00   | 0.99   | 0.98     | 0.98   | 0.98   | 1.00     | 0.91   | 0.95   | 0.95     | 0.98   | 0.96   | 0.96   | 54     |
| Person | 0.81     | 0.63   | 0.71   | 0.89     | 0.57   | 0.70   | 0.92     | 0.81   | 0.86   | 0.99     | 0.91   | 0.91   | 0.99     | 0.95   | 0.92   | 0.92   | 54     |
| Vehicle| 0.69     | 0.81   | 0.75   | 0.68     | 0.91   | 0.78   | 0.85     | 0.94   | 0.89   | 0.91     | 0.91   | 0.91   | 0.91     | 0.91   | 0.91   | 0.91   | 54     |
| Avg./Total | 0.81    | 0.81   | 0.81   | 0.83     | 0.83   | 0.83   | 0.91     | 0.91   | 0.91   | 0.91     | 0.91   | 0.91   | 0.91     | 0.91   | 0.91   | 0.91   | 162    |

the three metrics are significantly improved for all the cases, where the SURF-based approach with SVM provides the best performance with a value of up to 97% for precision, recall, and f-score. The best results for the other algorithms are given by KNN with ORB, NB with ORB, and DT with SURF, with up to 95%, 87%, and 89% for all metrics, respectively. For each descriptor, SVM outperforms the other algorithms in this second dataset. Focusing on the algorithm with the best performance, Fig. 4 shows the confusion matrices for this second dataset. Comparing this figure with Fig. 3, the confusion matrix shows that the use of generic classes, instead of specific ones, increases the detection capacity and the robustness of the machine learning-based ATR system.

Up to this point, we verified that the SURF-based approach with SVM provided a robust performance detecting three generic classes in LWIR images with targets of similar size. The idea of considering targets of similar size should not influence the performance of the system because the three descriptors used are size-independent. However, the low resolution of LWIR images could cause those targets which are far to the camera have a different appearance than expected. Despite the descriptors, the ML model is sensitive to these changes, so its detection capacity decreases. Following this idea, Table 4 shows precision, recall, and f-score for each pair descriptor-classification algorithm, while inferring about the dt_3_diff_ind database. Analyzing this table, the authors check that SVM with SURF and KNN with ORB provides a similar performance with up to 91% for the three metrics, where NB with SURF and DT with SURF provide up to 90% and 88%, respectively. Analyzing the confusion matrices of one of the algorithms providing the best performance, for instance, SVM in Fig. 5, the authors check that the confusion between classes was increased in comparison with Fig. 4, although it was decreased in comparison with Fig. 3. The main difference between them is the distance between the camera and the target, being the farthest the targets from dt_3_diff_ind database. This fact means that it could be interesting to constraint the identification of targets within a distance to avoid negatively affecting the detection capacity of the system.

B. EVALUATING THE FRAME-BASED PREDICTION METHOD IN DETECTION CAPACITY

This section analyzes the performance of the frame-based prediction method while using SURF, SIFT, and ORB descriptors and applying SVM, KNN, DT, and NB
classification algorithms. To this end, the authors consider a database combining individual targets and groups, i.e., the dt_6_eq_gr database. The reason for considering a database with a group of targets is because the authors are interested in analyzing the performance of the algorithm in a whole frame, which could include several targets. Note that the frame-based prediction method does not include a specific procedure for extracting and classifying ROIs; instead, the algorithm works with the whole frame directly.

Thus, Table 5 shows precision, recall, and f-score for each pair descriptor-classification algorithm, while inferring about the dt_6_eq_gr database. Analyzing this table, the authors check that SVM with ORB provides the best performance with up to 91% for the three metrics. The best results for the other algorithms are given by KNN with ORB, NB with SURF, and DT with ORB, with up to 83%, 77%, and 84% for the three metrics, respectively.

Analyzing the confusion matrices of the algorithm providing the best performance in Fig. 6, the authors check that the confusion between classes is slightly greater than before in Fig. 5. Thus, the confusion is reduced when not only individual targets are taken into account but also groups. When an image has a group of people, vehicle, or animals, bounding box might perform worse compared to the frame-based in those cases where targets are close together or they are near to the image border. The bounding box is limited by the ROIs which only focus on a single target. In this regard, we conclude that the frame-based create more robust system when multiple targets appear in the same frame and they can also be overlapped.

**C. EVALUATING THE PROPOSALS IN THE EDGE**

This section evaluates the two proposals by getting a trade-off between computing cost and detection capacity. To this end, the authors consider the workstation used before during the detection capacity analysis and a device with hardware that could be on the edge. The workstation has a 3.6 GHz Intel Core i7-7700 processor with 16 GB RAM, running a 64-bit Ubuntu operating system. The edge device is a PYNQ-Z1 platform by Xilinx. Although the PYNQ-Z1 device includes...
a Field Programmable System-on-Chip (FPSoC) combining a Field-Programmable Gate Array (FPGA) and a 650MHz dual-core Cortex-A9 processor, the authors only consider a single-core implementation of the proposals in this paper, running under a Linux kernel. That means that execution times could be reduced by applying parallel strategies, such as creating hardware accelerators in the FPGA for particular parts of the algorithm or using both cores of the processor.

Starting with the bounding box prediction method, Table 6 shows the average computing times (in seconds) obtained while inferring about the dt_3_eq_ind dataset for each pair descriptor-classification algorithm in the workstation and the PYNQ-Z1 device. The total average time (Total field) for a run is split into the time needed to get the feature descriptors (Descriptor field), to apply the BoF transformation (BoF field), and to predict using the supervised algorithm (ML field). As each of the frames in the dt_3_eq_ind dataset only includes an ROI, the ROI rate is calculated as one second divided by the total time, meaning the number of ROIs that can be calculated in one second. The speedup metric (Speedup field) is also shown, denoting the number of times that the implementation in the PYNQ-Z1 device is slower than the workstation. Analyzing this table, the authors reach that i) ORB needs the lowest computing time among the descriptors, ii) DT needs the lowest computing time among the classification algorithms, and iii) speedup is about 10.

Based on the computing times in Table 6 and the detection capacity analysis in Table 3, Figure 7 shows a trade-off between ROI rate (related to computing effort) and detection capacity for each pair descriptor-classification algorithm in the edge device. Note that the detection capacity is obtained as the average for the three metrics considered: precision,
TABLE 6. Computing time analysis applying the bounding box prediction method to dt_3_eq_ind.

| Descriptor | SVM | ORB | SURF | KNN | ORB | SURF | NB | ORB | SURF | DT | ORB | SURF |
|------------|-----|-----|------|-----|-----|------|----|-----|------|----|-----|------|
| Descriptor(s) | 0.0237 | 0.0104 | 0.0141 | 0.0320 | 0.0102 | 0.0184 | 0.0112 | 0.0106 | 0.0140 | 0.0152 | 0.0111 | 0.0149 |
| BoF(s) | 0.0052 | 0.0030 | 0.0042 | 0.0010 | 0.0031 | 0.0011 | 0.0010 | 0.0004 | 0.0011 | 0.0154 | 0.0033 | 0.0028 |
| MLoG(s) | 0.1305 | 0.0121 | 0.1247 | 0.0100 | 0.0146 | 0.0107 | 0.0005 | 0.0009 | 0.0005 | 0.0005 | 0.0007 | 0.0003 |
| Total | 0.1643 | 0.0115 | 0.1430 | 0.0431 | 0.0280 | 0.0301 | 0.0326 | 0.0152 | 0.0194 | 0.0444 | 0.0151 | 0.0231 |

ROI Rate: 6.0865, 63.6339, 6.9919, 23.2304, 35.7165, 33.1603, 30.6413, 65.9489, 51.1381, 20.6790, 66.2565, 43.2950
Frame Rate*: 1.5217, 15.8858, 1.7480, 5.8076, 8.9286, 8.2901, 7.4603, 16.4872, 12.7845, 5.1697, 16.5717, 10.8237

PYNQ-Z1 (single-core)

| Descriptor | SVM | ORB | SURF | KNN | ORB | SURF | NB | ORB | SURF | DT | ORB | SURF |
|------------|-----|-----|------|-----|-----|------|----|-----|------|----|-----|------|
| Descriptor(s) | 0.2554 | 0.0034 | 0.1275 | 0.2889 | 0.0223 | 0.1622 | 0.2415 | 0.0076 | 0.1621 | 0.2945 | 0.1003 | 0.1800 |
| BoF(s) | 0.0676 | 0.0437 | 0.0553 | 0.0134 | 0.0410 | 0.0140 | 0.0129 | 0.0448 | 0.0143 | 0.2011 | 0.0650 | 0.0367 |
| MLoG(s) | 1.3336 | 0.0230 | 1.7520 | 0.1025 | 0.2056 | 0.1900 | 0.0644 | 0.0129 | 0.0070 | 0.0097 | 0.0099 | 0.0088 |
| Total(s) | 2.1597 | 0.1462 | 1.9349 | 0.4429 | 0.3390 | 0.3002 | 0.3000 | 0.1553 | 0.1834 | 0.5003 | 0.1529 | 0.2215 |

ROI Rate: 0.4630, 6.0176, 0.5186, 2.2578, 2.9502, 3.0289, 3.3266, 6.4364, 5.4526, 1.9996, 6.5408, 4.5163
Frame Rate*: 0.1138, 1.5044, 0.1292, 0.5645, 0.7375, 0.7572, 0.8311, 1.6097, 1.3631, 0.4997, 1.6352, 1.1286

Speedup: 13.1460, 10.5415, 13.5283, 10.2890, 12.1058, 10.9479, 9.2167, 9.3787, 10.2427, 10.3466, 10.1343, 9.5907

FIGURE 7. ROI rate vs. detection capacity for each pair descriptor-classification applying the bounding box prediction method to dt_3_eq_ind in the PYNQ-Z1 device.

FIGURE 8. Frame rate vs. detection capacity for each pair descriptor-classification applying the frame-based prediction method to dt_6_eq_gr in the PYNQ-Z1 device.

f-score, and recall. The data in this figure is analyzed by following a Pareto theory strategy for a multi-objective problem, where both ROI rate and detection capacity are the objective functions to maximize and having two constraints defined by the application: detection capacity ≥ 85% and ROI rate ≥ 0.5 ROIs/s. As a result, the non-dominated Pareto front is composed of three solutions with the same relevance. They are from lower to higher detection capacity and from higher to lower ROI rate as follows: ORB-NB, ORB-SVM, and SURF-SVM.

Following with the frame-based prediction method, Table 7 shows the average computing times obtained while inferring about the dt_6_eq_gr dataset for each pair descriptor-classification algorithm in the workstation and the edge device. In this case, as the frames in the dataset could include several targets and the frame-based prediction method infers without extracting ROIs, working directly with the whole frame, the authors opted for using the frame rate metric instead of the ROI rate, meaning the number of frames that can be calculated in one second. Analyzing this table, the authors reach the same conclusions as before, that is i) ORB needs the lowest computing time among the descriptors, ii) DT needs the lowest computing time among the classification algorithms, and iii) speedup is about 10.

Based on the computing times in Table 7 and the detection capacity analysis in Table 5, Fig. 8 shows a trade-off between frame-rate and detection capacity for each pair descriptor-classification algorithm in the edge device. As before, the data is analyzed following a Pareto theory strategy, where both frame rate and detection capacity are the objective functions to maximize, having the constraints: detection capacity ≥ 85% and frame rate ≥ 0.5 Frames/s. Thus, the non-dominated Pareto front is composed of two solutions with the same relevance. They are from lower to higher detection capacity and from higher to lower frame rate as follows: ORB-DT and ORB-SVM.
TABLE 7. Computing time analysis applying the frame-based prediction method to dt_6_eq_gr.

| Descriptor   | SVM | KNN | NB | DT |
|--------------|-----|-----|----|----|
| Time (s)     |     |     |    |    |
| Frame Rate   |     |     |    |    |
| Speedup      |     |     |    |    |

Comparing both frame rates in Tables 6 and 7, the authors reach that the frame rate metric is significantly higher when applying the frame-based model, as expected because there is no need of iterating over ROIs. From this analysis, when multiple targets appear inside the image, on average, the bounding box requires high computational time as it focuses on one target per ROI. In the edge, for each descriptor and classification algorithms, the frame-based achieves up to 26 times higher frame rate than bounding box, that is, the frame-based approach decreases the computing cost 26 times, as they use the same resources.

Attending to the approaches, the descriptors computing cost is independent of descriptor selected. Also, it is independent of the image size, as can be shown in between the time required to process a particular ROI and the whole frame, as shown in tables 6 and 7. The BoF model computing cost depends on descriptor used, but the approach. Furthermore, in terms of computing time, the BoF calculation is negligible compared to the descriptor calculation. Tables 6 and 7 also show that the computing time for classification algorithms in the bounding box is lesser than in the frame-based. As the computational effort for a particular descriptor is the same regardless of the approach, it impacts the information that the descriptor can gather from the image. Then, the frame-based detection method might gather less information than the bounding box. Hence, the classification algorithm for the frame-based achieves more complexity, increasing the classification computing cost. However, in some cases the classification algorithm its computing cost, in terms of computing time, is sufficiently low that can be considered negligible from the whole computing time of the approach.

In this regard, the descriptor calculation has the highest impact on the solutions according to the frame rate. Thus, the highest frame rates shown in the Pareto front analysis are achieved by both approaches are those where the ORB descriptor is used, as it is the most lightweight descriptor used. Despite the classification algorithm has a stronger impact on the frame-based computational cost, the computational cost does not impact as the descriptor one. As a result, the computational cost approach is lesser than the bounding box. So, we conclude that the ORB-SVM frame-based approach fits better to our use case as it provides more robustness, high frame rate and a detection capacity of 91% in the machine learning-based ATR system.

D. THE PROPOSALS AND THE STATE-OF-THE-ART

Up to this point, the authors reached that the bounding box prediction method provided a detection capacity from 87% to 97% with a frame rate from 0.13 to 1.61 running on the edge device and a frame rate from 1.75 to 16.49 running on the workstation. For the frame-based prediction method, the detection capacity obtained was from 84% to 91% with a frame rate from 3.77 to 5.71 running on the edge device and a frame rate from 44.06 to 59.17 running on the workstation. Comparing such results with the works previously discussed in Table 1, the authors reach that:

- The two proposals achieved detection capacity metrics in line with the state-of-the-art in ATR systems based on IR imagery, being the highest detection capacity
metric obtained a value of 97%. This value is substantially above the average detection capacity for the works in Table 1, which is 90.33%. Note that most of the works analyzed provided the accuracy metric as a detection capacity metric instead of using more confident metrics, such as the ones considered in this paper. Therefore, the detection capacity analysis in these works could be biased. Focusing on the works providing higher detection capacity metrics than ours (97%) [33], [38], [42], the authors reach that two of the works [38], [42] were based on CNNs whose approach might not be applied in edge platforms due to its resource utilization, such as convolutional layers with 256 filters.

- Regarding the computational cost, only the works in [32], [35], [39], [40], [42] provided frame rates running in a workstation. For the case of [40], [42], the frame rates shown in these works are comparable with the frame rates from our proposals running on the workstation, which is relevant because they considered powerful GPUs. For the case of [39], the algorithm obtained a frame rate of 200 but in a GPU and with lower detection capacity than ours. For the case of [35], the algorithm obtained a lower detection capacity and frame rate (0.16) than our approaches. For the case of [32], the frame rate is comparable, but our approach provides a higher detection capacity when considering a high frame rate. From the analysis of these works, we reach that the computational cost of the proposals is within the state-of-the-art.

- Regarding the computational cost in the edge, only the works in [37], [41] provided frame rates. For the case of [37], they obtained a higher frame rate, but the detection capacity was significantly lower, which might not be suitable for certain applications, such as surveillance. For the case of [41], the detection capacity was similar to ours, but the frame rate was significantly lower, approximately eighteen times lower than our proposal for similar accuracy. From the analysis of these two works and according to the literature introduced before for other implementations in the edge with frame rates from 0.4 to 14 [46]–[48], we reach that the computational cost for both proposals in the edge is in line with the state-of-the-art.

Note that the comparison with previous works is limited because i) the problem addressed was not solved before in the literature, ii) the hardware used in the workstation is different in each work, and iii) the hardware used in the edge is different in each work.

VI. FINAL REMARKS

As a result of evaluating both prediction methods, the authors concluded that in a multiclass scenario, i) the usage of generic classes is recommended because of the low resolution of IR images, ii) a solution which tries to achieve high detection capacity might focus on the SVM classifier, iii) a solution which tries to achieve high frame rate values might focus on the ORB descriptor, iv) the solution providing the best performance in terms of detection capacity was SURF-SVM bounding box prediction approach, with up to 97%, and v) the solution with highest frame rate was ORB-DT frame-based prediction method, with up to 59.17 frames/s in the workstation and 5.70 frames/s in the edge device.

Focusing on a classification problem with three classes (animal, vehicle, and person), as the one shown in dt_3_eq_ind, we concluded that vi) the confusion between classes decreased using generic categories (such as animals), instead of identifying a specific target (such as camel), meaning an increase in detection capacity. Furthermore, when multiple targets appeared and were overlapped in the image, as the one shown in dt_3_diff_ind, vii) the detection capacity is negatively affected because of the low resolution of IR images. Thus, the authors concluded that viii) in that scenario combining generic individual and group targets, as the one shown in dt_6_eq_gr, increases the performance, but this method can only be applied in the frame-based prediction platform. In terms of detection capacity, on the one hand, ix) in scenarios where only one target appears in the image SURF-SVM bounding box prediction might be applied. On the other hand, x) in multiple target scenarios the ORB-SVM frame-based prediction method might be applied to achieve high frame rate and a detection capacity above 90%.

Evaluating the method in the edge, the authors concluded that xi) ORB requires the lowest computing time among the descriptors and it has more impact in the machine learning-based ATR system than the BoF and classification methods, xii) DT needs the lowest computing time among the classification algorithms, and xiii) the edge device is 10 times slower than the workstation. However, xiv) the solutions from both non-dominated Pareto fronts achieve the state-of-the-art frame rate and detection capacity for the edge domain, thus, being suitable for the desert border monitoring use case.

Future lines of research could include obtaining real traces of energy consumption to propose custom optimizations. Moreover, it could be interesting to propose parallel strategies to be implemented the proposals in low-power multicore microprocessors and FPGAs.

ACKNOWLEDGMENT

The authors would like to thank Mr. Azrouk Abdel Latif, who recently served as Cultural Attaché at the Embassy of Libya in Madrid, providing administrative support that greatly helped the research.

REFERENCES

[1] M. S. Jadin and K. H. Ghazali, “Gas leakage detection using thermal imaging technique,” in Proc. 16th Int. Conf. Comput. Modeling Simulation, Mar. 2014, pp. 302–306.

[2] C. Cica and L. Filipoaia, “Sustainability enabling research methods for development of integrated border security systems,” in Proc. 9th Int. Conf. Electron., Comput. Artif. Intell. (ECAI), Jun. 2017, pp. 1–4.

[3] E. F. Gautier, Sahara: The Great Desert. Evanston, IL, USA: Routledge, 2019.
[51] Z. Xu, D. Zhang, and L. Du, “Moving object detection based on improved three frame difference and background subtraction,” in Proc. Int. Conf. Ind. Inform.-Comput. Technol., Intell. Technol., Ind. Inf. Integr. (ICIICI), Jan. 2017, pp. 79–82.

[52] S. Jeannin and M. Bober. (1999). Shape Data for the MPEG-7 Core Experiment CE-Shape-1 Online. [Online]. Available: http://www.cis.temple.edu/latecki/TestData/mpeg7shapeB.tar.gz

[53] J. Wang, H. Jiang, Z. Yuan, M.-M. Cheng, X. Hu, and N. Zheng, “Salient object detection: A discriminative regional feature integration approach,” Int. J. Comput. Vis., vol. 123, no. 2, pp. 251–268, Jun. 2017.

[54] S.-H. Lee, J.-H. Kim, K. P. Choi, J.-Y. Sim, and C.-S. Kim, “Video saliency detection based on spatiotemporal feature learning,” in Proc. IEEE Int. Conf. Image Process. (ICIP), Oct. 2014, pp. 1120–1124.

KHALIFA M. BELLAZI received the M.Sc. degree in telecommunication and computer networking from London South Bank University, U.K., in 2004. He is currently pursuing the Ph.D. degree with the Department of Electrical and Electronic Engineering, Universidad Politécnica de Madrid. His research interests include edge computing, object-recognition, and automated targets recognition.

RODRIGO MARINO (Graduate Student Member, IEEE) received the B.Sc. degree in industrial electronics engineering and automation and the M.Sc. degree in industrial engineering from the Universidad de Vigo, Spain, in 2015 and 2017, respectively. He is currently pursuing the Ph.D. degree in industrial electronics with the Universidad Politécnica de Madrid. He is participating in a national research project, PLATINO, related to the enhance the acquisition systems, combining machine learning techniques in embedded systems, in order to develop expert sensors for the agro-food industry. He has also participated in an industrial project, REMO, with Indra and Repsol companies, in order to create a framework for chemical detection. His current research area is in the field of machine learning applied to embedded systems also known as expert embedded computing.

JOSE M. LANZA-GUTIERREZ received the B.S. and M.S. degrees in computer science and the master’s degree in grid computing and parallelism from the University of Extremadura, Caceres, Spain, in 2008, 2009, and 2010, respectively, and the Ph.D. degree in computer science from the University of Extremadura under the guidance of Prof. Dr. Juan A. Gomez-Pulido in 2015. He is currently an Assistant Professor with the University of Alcalá, Alcalá de Henares, Spain. He has authored or coauthored more than 40 publications, including Journal Citation Report articles in journals, such as Applied Soft Computing, Expert Systems with Application, BMC Bioinformatics, Soft Computing, Reliability Engineering and System Safety, IEEE Access, and the IEEE INTERNET OF THINGS. His main research interests include metaheuristics, the Internet of Things, digital embedded systems, and machine learning.

TERESA RIESGO (Member, IEEE) received the M.Sc. and Ph.D. degrees in electrical engineering from the Universidad Politécnica de Madrid (UPM), Madrid, Spain, in 1989 and 1996, respectively. Since 2003, she has been a Full Professor of electronics with UPM. Her research interests are focused on embedded system design, wireless sensor networks, the Internet of Things, configurable systems, and power estimation in digital systems. She has participated and acted as the main researcher in more than 60 national and European research, development and innovation projects, almost always in collaboration with the industry. As a result of her research activity, she has authored or coauthored more than 200 scientific documents in journals and international congress proceedings. She has been the Director of the Center for Industrial Electronics, the Deputy Director of Research and International Relations with ETSII, UPM. She was the Director of the Department of Automation, Electrical and Electronic Engineering and Industrial Computing, UPM, until 2018. Since 2003, she has been the Representative of the UPM in the Top Industrial Managers in Europe network and a member of its management committee. In 2018, she has been appointed the General Director of Research, Development and Innovation, a position under the Ministry of Science, Innovation and Universities of the Government of Spain. She participates regularly in evaluation panels of national projects and EU framework programs.

* * *

K. M. Bellazi et al.: Toward an Machine Learning-Based Edge Computing Oriented Monitoring System