Neural Computing and Applications
https://doi.org/10.1007/s00521-023-08741-4

ORIGINAL ARTICLE

Object detection in traffic videos: an optimized approach using super-resolution and maximal clique algorithm

Iván García-Aguilar[1,2] · Jorge García-González[1,2] · Rafael Marcos Luque-Baena[1,2] · Ezequiel López-Rubio[1,2]

Received: 21 January 2023 / Accepted: 31 May 2023

Abstract

Detection of small objects is one of the main challenges to be improved in deep learning, mainly due to the small number of pixels and scene's context, leading to a loss in performance. In this paper, we present an optimized approach based on deep object detection models that allow the detection of a higher number of elements and improve the score obtained for their class inference. The main advantage of the presented methodology is that it is not necessary to modify the internal structure of the selected convolutional neural network model or re-training for a specific scene. Our proposal is based on detecting initial regions to generate several sub-images using super-resolution (SR) techniques, increasing the number of pixels of the elements, and re-infer over these areas using the same pre-trained model. A reduced set of windows is calculated in the super-resolved image by analyzing a computed graph that describes the distances among the preliminary object detections. This analysis is done by finding maximal cliques on it. This way, the number of windows to be examined is diminished, significantly speeding up the detection process. This framework has been successfully tested on real traffic sequences obtained from the U.S. Department of Transportation. An increase of up to 44.6% is achieved, going from an average detection rate for the EfficientDet D4 model of 14.5% compared to 59.1% using the methodology presented for the first sequence. Qualitative experiments have also been performed over the Cityscapes and VisDrone datasets.

Keywords Convolutional neural networks · Super-resolution · Test time augmentation · Object detection · Small objects

1 Introduction

In recent years, object detection has been applied to many environments, including autonomous driving and video surveillance. Numerous video surveillance systems in road networks offer the potential to use and evaluate the gathered information to identify significant events. Therefore, it is necessary to get reliable object detection for the sequences captured by these systems, constituting one of the main problems in computer vision. This task was performed through classical techniques. Today, many advances have been established within the area of deep learning. The performance in classifying and detecting objects has drastically improved thanks to convolutional neural networks over the last few years. The area of highway surveillance systems is an excellent application for this type of technology. Information on traffic density, road safety, and pollution estimation could be obtained by analyzing their direction, speed, and behavior. There are many pre-trained models available for detecting and locating elements from images. However, even with recent advancements, some object detection-related challenges still need to be solved for small object detection. In this context, the vehicles are frequently smaller than the overall

✉ Iván García-Aguilar
ivangarcia@lcc.uma.es

Jorge García-González
jorgegarcia@lcc.uma.es

Rafael Marcos Luque-Baena
rmluque@lcc.uma.es

Ezequiel López-Rubio
ezeqlr@lcc.uma.es

[1] Department of Computer Languages and Computer Science, University of Málaga, Bulevar Louis Pasteur, 35, 29071 Málaga, Spain

[2] Biomedical Research Institute of Málaga (IBIMA), C/ Doctor Miguel Díaz Recio, 28, 29010 Málaga, Spain

Published online: 15 June 2023

image size because of the camera distance. As a result of budget limitations, cameras are often old-fashioned or have poor performance, and image resolution is usually low, so each vehicle is composed of a small number of pixels with simple shapes and not small parts detail.

The angle, weather, and lighting conditions are variable since these systems must be able to operate in different locations and situations. It is vital to use a detection system that is as accurate as possible to ensure a smooth and efficient tracking process. One must also consider the computational resources required to identify elements within a video sequence. The use of graphics processing units (GPU) is a crucial factor in applying the method in a reasonable time. Works such as the one proposed by Lingzhi Shen et al. [1] present a YOLOv3-based method to enhance the capability of cross-scale detection and focus on the valuable area, reducing the complexity of training and paving the way for fast convergence. Zhihe Zhuang et al. [2] proposed optimal iterative learning control *ILC* algorithm addresses nonuniform trial lengths and input constraints, offering potential improvements. Zhou et al. [3] propose a PD-type iterative learning control algorithm for spatially interconnected systems with unstructured uncertainty that have the potential to be applied in that field. Several algorithms focused on minimizing the computation times required by the object detection model have been developed, such as [4–6]. Despite the progress in adopting these methods, such as [7], no solution trivializes the computation time required to apply the object detection model.

Identifying objects of small size is just as crucial as identifying medium and large objects. The method presented has a wide range of relevant uses. This paper focuses primarily on improving object detection algorithms used for road sequences. The proposed solution can be applied whenever small elements are not initially detected in their entirety. Other applicable contexts are the industrial field or the medical area. Under the premises described above, there are significant shortcomings in detecting small-scale elements due to the lack of methods and techniques to improve performance and the low average detection accuracy established by the pre-trained models.

The proposed solution in this article is a meta-model that optimizes the performance of pre-trained convolutional neural networks to improve small-sized object detection using super-resolution (SR), increasing the class score to obtain more reliable detections. This process avoids modifying the structure or re-training existing models, which are already pre-trained. A set of regional proposals is determined based on the tentative elements initially detected and a group of fixed areas. These regions will be processed with a super-resolution model, generating new sub-images selected through an optimization process on which the model must re-infer, improving the time required to process sequences. For this purpose, a graph is calculated to generate the optimal number of sub-images according to the initial elements detected based on a window's size. We first studied the optimal window size and performed detailed experiments to test the effectiveness of the presented proposal. Subsequently, we prove quantitatively and qualitatively that our methodology improves the performance initially obtained by the raw pre-trained model.

The rest of this article is organized as follows. Section 2 on page 2 sets out the related work. Throughout Sect. 3 on page 3, the improvements developed are detailed, explaining the implemented workflow. Section 4 on page 4 includes the study about the windows-sliding (R) and their respective results. Finally, in Sect. 5 on page 5, the conclusions and the future works to be developed according to the proposed solution are outlined.

2 Related work

In this section, we first introduce deep learning-based general object detectors, and then discuss relevant small object detection methods, the advances in convolutional neural network (CNN) models for super-resolution applications, and the contributions of our proposal.

2.1 Convolutional neural networks for object detections

According to developments in deep learning, it has been shown that approaches based on convolutional neural networks (CNN) have considerably improved object classification and detection, thus obtaining good results. In line with this improvement, several pre-trained models are available. These can be classified into two main groups. The first is R-CNN (region-based convolutional neural network), which comprises two stages. First, it identifies the areas of interest given an image through a selective search or by using a network that proposes regions. Subsequently, the model will infer over these areas to detect elements. Several advances and improvements have been made in this group, giving rise to models such as *Faster R-CNN* [8]. This model introduces the region proposal network (RPN) concept. This fully convolutional network predicts the score and objects boundaries. These proposals will be inferred using the element detection domain model. Another model that stands out in this field is the one known as *EfficientNet* [9]. This model consists of a convolutional neural network architecture applied in combination with a scaling method. Uniform scaling of the dimensions determines the compound coefficient regarding depth, width, and resolution. A base network has been developed by searching for the optimal architecture that composes the model, which is scaled to obtain a family of models. The most simple one is called B0, which is based on the inverted bottleneck residual blocks of *MobileNetV2* [10], to the most complex one set as B7. The models presented above focus on the accuracy in detecting the elements present in the image given as input, ignoring the speed required for processing. For real-time detection, one-step methods are available. Therefore, they ignore the generation of regional proposals and use local information. In this area, highlight models such as *SSD* [11] or *YOLO* [12]. According to the last model, several advances, such as the one established in [13], integrate the convolutional block attention model (CBAM) to find the specific region in scenarios with several dense objects. In addition, strategies based on data augmentation, multiscale testing, and an additional classifier are applied.

Works such as the one proposed by Subudhi et al. [14] present a new algorithm for detecting and tracking moving objects. A Markov random field (MRF) model is first used to identify the scene attributes and to obtain a spatiotemporal segmentation of the elements. This segmentation uses the maximum a posteriori probability (MAP) estimation technique and a heuristic to optimize the required time. In the field of element detection and tracking, there are works such as the one proposed by Travis Mandel et al. [15], where an algorithm called robust confidence tracking (RCT) designed to improve element tracking through accurate values of confidence in detection is established, thus obtaining a robust performance. In video surveillance, Kavitha et al. [16] propose an extreme machine learning and action recognition scheme developed for semantic concept detection in unnatural videos called MLE. From video surveillance sequences, an efficient scheme is provided by encoding features with the help of a locally aggregated descriptor vector (LAADV) to reduce the required computational time. This paper uses cliques through the modified branch-and-bound method (MBBM) to solve such a problem. With the resulting features, deeper features are obtained using CNN, thus demonstrating that the proposed technique offers higher accuracy and lower time complexity. Sheng Ren et al. [17] introduced an advanced super-resolution framework for video object detection. The framework combines object detection algorithms, video keyframe selection algorithms, and super-resolution reconstruction algorithms. The proposed deep learning-based intelligent video detection object super-resolution (SR) method uses a regression-based object detection algorithm, a key video frame selection algorithm, and an asymmetric depth recursive back-projection network for super-resolution reconstruction. This approach enhances the resolution and visual clarity of key objects, improving the accuracy and effectiveness of object detection in videos.

2.2 Advances in the field of small object detections

Despite the pre-trained models available, it has significant problems to be improved. Their accuracy rate drops considerably when the size of the elements is not big enough. They are composed of several layers that process the image, performing a series of convolutions. In each phase, the number of pixels of the image given as input is reduced, causing a loss in detecting small elements. In addition, they have been trained and evaluated on well-known datasets such as *ImageNet* [18] or *MS COCO* [19], in which most of the objects contained in them have large portions concerning the image. For example, *EfficientDet* [20], which has a high rate for medium-sized elements, its efficiency decreases with smaller elements. Models such as *YoloV4* [12] obtain an overall mean average precision (mAP) of 43%. However, this score drops to 24.3% for small objects.

In small object detection, several advances, such as the one defined by Rabbi et al. [21], propose a new architecture formed by three components. An edge-enhanced super-resolution GAN (EESRGAN) is applied in combination with an edge-enhancement network (EEN) and an object detection model. Through the application of the GAN network, the quality of the image given as input is improved. For this purpose, they used different end-to-end detector networks where the loss was backpropagated into the EESRGAN to improve the detection performance. This advance requires training to be applied in a specific scene. However, in our proposal, the model does not need re-training. Other works, such as the one by Deng et al. [22], propose an extended feature pyramid network (EFPN) with an extra high-resolution pyramid level specialized for small object detection. For this purpose, they developed a new module called feature texture transfer (FTP) to apply super-resolution and extract credible regional details simultaneously. The main difference with our proposal is that the layers that compose the object detection model must be modified. Our proposal can be directly applied to pre-trained models, avoiding modification. For the autonomous driving and vehicle detection domain, Su et al. [23] propose a feature pyramid spatial attention (FPSA), which uses high-level features as attention information according to low-level ones. Other works, such as Khan et al. [24], propose a two-step-based approach. It applies the *Faster R-CNN* model [8] to detect the given vehicles in an image and subsequently employs morphological operations to reduce those regions that are not of interest.

2.3 Advances in super-resolution

According to the super-resolution (SR) application, several models are available for direct use, such as [25–28]. Each of these has a specific structure. Since we want to optimize the times required to improve initial detections, we have selected *fast super-resolution convolutional neural network (FSRCNN)* [26]. This model introduces a deconvolution layer to perform upsampling at the network's end. Three steps in FSRCNN replace the nonlinear mapping step in *super-resolution convolutional neural network (SRCNN)*: shrinking, mapping, and expanding. Finally, the smaller filter sizes and a deeper network structure provide better performance and are tens of times faster than other models. In particular, *FSRCNN-s* can be implemented in real time on a generic CPU.

Applied to sequences, works like the one presented by Kong et al. [29] propose a method to enhance the spatial resolution in video sequences by combining information with different spatiotemporal resolutions from various cameras. This is achieved by constructing a training dictionary using high-resolution images captured by a still camera and enhancing low-resolution video by searching this scene-specific database. This approach generates more realistic results because the training is based on the specific scene. Finally, the method presented enforces spatiotemporal constraints using conditional random fields (CRF). The problem of video super-resolution is framed as finding the high-resolution video that maximizes the conditional probability. Camargo et al. [30] present a framework that does not require the construction of sparse matrices. This approach utilizes image operators in the spatial domain and an iterated back-projection method to produce super-resolution mosaics from frames of surveillance video captured by unmanned aerial systems (UAS), where the information analysis is usually affected by different factors, such as motion blur. Numerical methods such as the steepest descent, conjugate gradient, and Levenberg–Marquardt algorithm were employed to solve the nonlinear optimization problem in modeling the super-resolution mosaic.

2.4 Contributions of our proposal

Our proposal focuses on improving small vehicle detections on urban roads and aims to optimize the framework [31]. This framework improves the mean accuracy precision (mAP) by re-inferring on multiple sub-images generated by applying super-resolution. Therefore, a new sub-image is generated for each initially detected element. Since we are analyzing sequences with elements close to each other, several similar sub-images are generated. This fact substantially increases the time required to process the sequence. For this purpose, a new module is introduced for calculating the minimum clique list, maximizing the number of elements. Subsequently, a greedy algorithm is applied to remove similar images, thus improving the total time required by the object detection model. Our proposal finds new visual elements thanks to the super-resolution processes, the new enhancements added, and the further optimization stage presented.

The presented methodology provides an efficient solution to improve the detection and precision of elements in images. Compared to previously described approaches, creating super-resolved images allows, first of all, to significantly improve the quality of the images, thus improving the accuracy in those cases where the system generates low-quality images. Therefore, the re-inference on these areas using convolutional neural networks avoids the re-training of the model and the modification of its structure. Thus, the described methodology can be implemented in any convolutional object detection model, making it versatile and adaptable for different scenarios and applications. This article discusses the implementation of the same in road sequences. However, this methodology can be applied in other areas, with a significant impact on the quality of the results, such as detecting elements in medical images and remote sensing and video surveillance systems, among others.

3 Methodology

This section details the presented methodology in Fig. 1, stating each of the steps that compose it. We take advantage of the fact that detection models based on convolutional neural networks perform well in the high-resolution domain. Given an input video frame $\mathbf{X}_{\mathrm{LR}}$ of size $W \times H$ pixels where W is the width and H the height of the frame, the first step is applying a super-resolution network $\mathcal{G}$ to the original low-resolution image $\mathbf{X}_{\mathrm{LR}}$ to obtain a high-resolution version $\tilde{\mathbf{X}}_{\mathrm{HR}}$:

$$\tilde{\mathbf{X}}_{\mathrm{HR}} = \mathcal{G}(\mathbf{X}_{\mathrm{LR}}) \tag{1}$$

Some pre-trained models are denoted as $\mathcal{G}$ for the execution of super-resolution (SR) processes to increase the initial resolution of an image. Our goal is to process the images optimally, so we have selected the model denominated as fast super-resolution convolutional neural network (FSRCNN) [26] for being one of the fastest models. This model has several versions available for use. Each of them determines a particular upscaling factor Z ($X2$, $X3$, $X4$). The proposal can be performed by adjusting the upscaling factor. It is crucial to determine the context of the scene when choosing a specific Z to ensure accuracy.

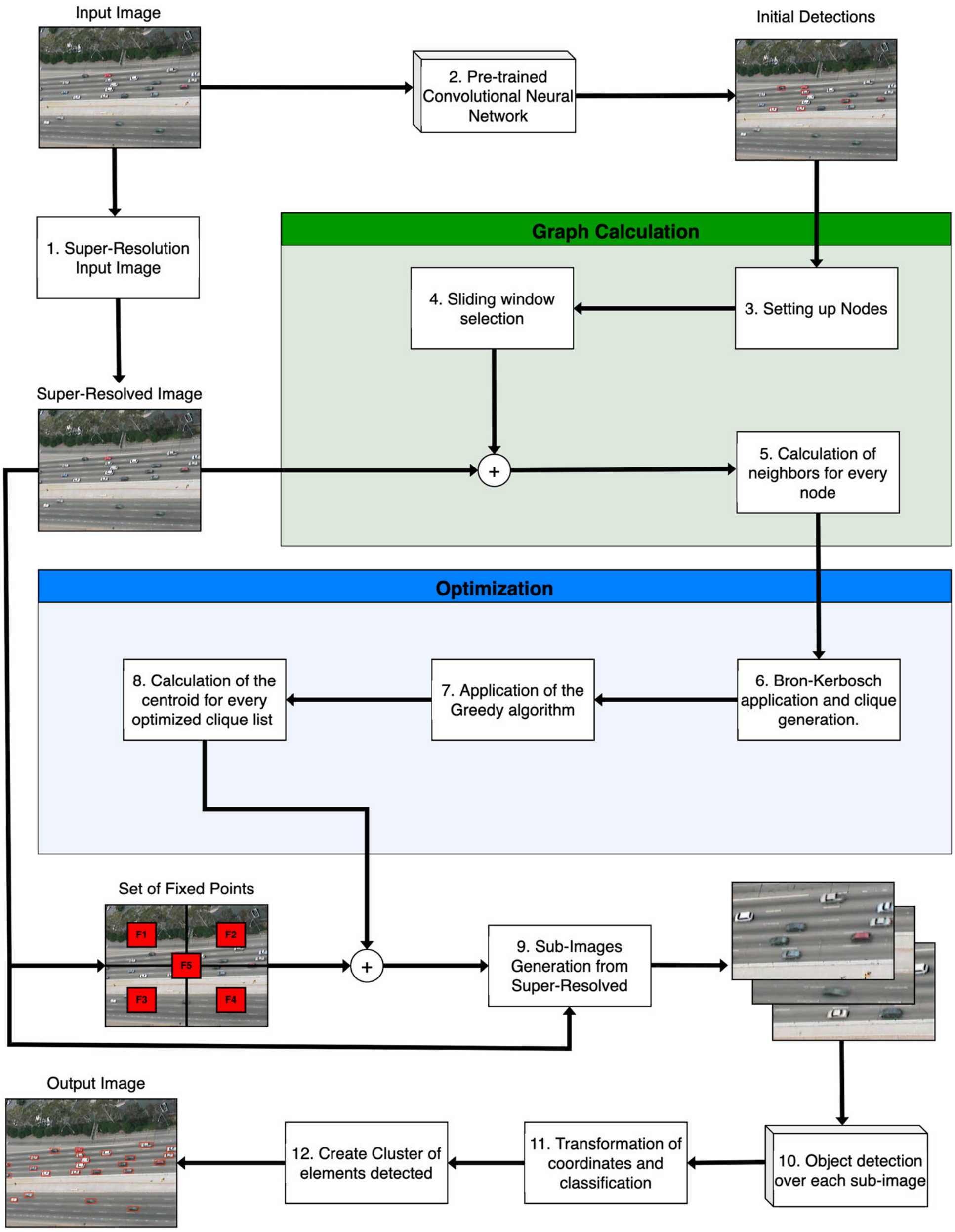
Input Image
2. Pre-trained Convolutional Neural Network
Initial Detections
1. Super-Resolution Input Image
Graph Calculation
4. Sliding window selection
3. Setting up Nodes
Super-Resolved Image
+
5. Calculation of neighbors for every node
Optimization
8. Calculation of the centroid for every optimized clique list
7. Application of the Greedy algorithm
6. Bron-Kerbosch application and clique generation.
Set of Fixed Points
F1
F2
F5
F3
F4
+
9. Sub-Images Generation from Super-Resolved
Output Image
12. Create Cluster of elements detected
11. Transformation of coordinates and classification
10. Object detection over each sub-image

Fig. 1 Workflow of the proposed technique

However, using a very high upscaling factor Z may cause the object detection model to classify the object due to the enlarged size incorrectly. Let $(WZ) \times (HZ)$ pixels be the size of the super-resolved image $\tilde{\mathbf{X}}_{\mathrm{HR}}$, where Z is the upscaling factor ($X2$ selected).

The second step in Fig. 1 consists of processing the input image $\mathbf{X}_{\mathrm{LR}}$, which has a low resolution with the convolutional neural network model $\mathcal{F}$ to yield a set of tentative detections D_{LR}:

$$D_{\mathrm{LR}} = \mathcal{F}(\mathbf{X}_{\mathrm{LR}}) \tag{2}$$

This gives a list of detections named D_{LR}:

$$\mathcal{D}_{\mathcal{LR}} = \{(x_i, y_i, h_i, w_i, l_i, s_i) \parallel i \in \{1, ..., N\}\} \tag{3}$$

where (x_i, y_i) are the coordinates of the upper left corner of the ith detection, h_i is its height, w_i is its width, l_i is the class label, s_i is the classification score (detection confidence), and N is the number of initial elements detected.

The list of detections D_{LR} is then filtered by a threshold T because the class score s_i obtained for some of them will be low. We define those detections as high confidence if they overreach T, set to 0.35, or low confidence if they do not exceed this threshold. This threshold represents the minimum confidence that an object must have to be considered as a positive detection, and it takes a value from 0 to 1. Therefore, any element with a confidence score lower than 0.35 will be ignored as a false positive. This threshold has been selected according to the dataset's nature and the complexity of the objects to be detected on which the presented methodology is applied. By setting a threshold higher than 0.35, it is possible to increase the specificity and thus reduce the false positive rate while omitting potential false negatives. With a lower threshold, sensitivity is increased, but the risk of introducing false positives increases. The dataset is composed of small elements, difficult to detect due to their size and the diffusion they present with the background, among other problems. Therefore, the threshold established to carry out the experimental phase in section 4 represents a reasonable compromise between accuracy and specificity over the scope. After filtering out the detection with a confidence lower than a threshold T, we obtain a reduced set of high confidence detections $\mathcal{D}'_{\mathrm{LR}}$:

$$\mathcal{D}'_{\mathcal{LR}} = \{(x_i, y_i, h_i, w_i, l_i, s_i) \in \mathcal{D}_{\mathcal{LR}} \parallel s_i > T\} \tag{4}$$

Let us note D'_{LR}, the cardinal of $\mathcal{D}_{\mathcal{LR}}$, i.e., the number of high confidence detections in D'_{LR}. Also, let us consider a ratio parameter R, which the user can edit to define some specific windows of size $(WZR) \times (HZR)$ pixels in the super-resolved image $\tilde{\mathbf{X}}_{\mathrm{HR}}$ to perform a second pass of the object detection network.

The aim is to detect more objects and increase confidence in the already found detections while keeping a low computational load. This way, we need to perform object detection in the sub-images $\mathbf{I}'$ generated from the super-resolved image $\tilde{\mathbf{X}}_{\mathrm{HR}}$ without having to pass the object detection network exhaustively on all possible windows that could be defined. Therefore, to set the sub-images $\mathbf{I}'$ to be generated to re-infer, we build a graph $\mathcal{GR}$ with D'_{LR} nodes that represent the high confidence detections. Two nodes i and j are connected in $\mathcal{GR}$ if and only if they correspond to two detections that would fit into the same window W of size $(WZR) \times (HZR)$ pixels. Steps 3, 4, and 5 in Fig. 2 show the graph generation diagram $\mathcal{GR}$.

Next, in step 6, a maximal clique search by the Bron–Kerbosch algorithm is done on $\mathcal{GR}$, so that all maximal cliques $C_k \subseteq \mathcal{D}'_{LR}$ are found. Let us note $\mathcal{C}$, the set of all maximal cliques of $\mathcal{GR}$, and K, the number of such cliques. A maximal clique is a set of nodes such that all of them are connected with each other and that no other clique includes it. It must be highlighted that all the detections that belong to the same clique could be processed at once by selecting a single window of size $(WZR) \times (HZR)$ pixels that includes all of them. By generating the maximum list of cliques, it is possible to determine which parts of the image the elements are most frequently concentrated. This list will be used to generate the optimal sub-images. This mainly prevents the detection model from re-inferring on similar sub-images, optimizing the time required to apply our proposal. However, there are cases in which certain cliques are contained in others with a greater number of elements. Therefore, an optimization algorithm has been developed and is described below.

The greedy algorithm in step 7 is employed to find out a set of maximal cliques that covers all the detections in $\mathcal{D}'_{LR}$. At least P cliques select each detection. P is a tunable parameter that the user can previously define. This value sets the number of times a particular element will be considered when generating the optimal cliques C' list. The algorithm reads as follows:

1. Set an integer count for each node of $\mathcal{GR}$ to zero.
2. Set a Boolean flag for each node of $\mathcal{GR}$ to false.
3. Set a Boolean flag for each maximal clique of $\mathcal{C}$ to false.
4. Find a maximal clique in $\mathcal{C}$ that is flagged as false, with the highest number of false-flagged nodes. Let us note C' such maximal clique.
5. Set the flag of C' to true to indicate that it has already been selected.
6. Increase the counter of all the nodes in C' by one. Set the flag of all nodes whose counters have reached P to true to indicate that they have already been covered. When one of the elements has been included at least P times in several cliques, it is removed from the rest of the cliques to be validated.
7. Those cliques contained in others with more elements are removed.
8. If all the nodes of $\mathcal{GR}$ are flagged as true, then halt. Otherwise, go to step 4.

After the execution of this greedy algorithm, an optimal list of cliques C' will have been established. According to the optimized clique list C', the smallest sub-images that maximize the elements contained in each will be generated. For each of the optimal cliques obtained, a new sub-image is generated on which the object detection model will infer again. Let C'_i be one of the cliques generated after performing the above steps. It will be formed by a set of N nodes, where each node represents one detected element.

$$C'_i = \{(x_i, y_i, h_i, w_i, l_i, s_i) \parallel i \in \{1, ..., N\}\} \tag{5}$$

In step 8, the center of each sub-image to be generated, denoted as $\mathbf{I}'_k$ is calculated. Therefore, the centroid (c_x,c_y) of the elements contained in C'_i is computed. According to a width w_n and a length h_n, which are initially set to the size of the initial image given as input $\mathbf{X}_{\mathrm{LR}}$, The window $\mathbf{X}_i$ is centered at the center of the detection (c_x,c_y) and the new sub-image $\mathbf{I}'_k$ will be created starting from the super-resolved image $\tilde{\mathbf{X}}_{\mathrm{HR}}$ in step 9. We also create five sub-images due to a set of fixed zones given the input image, denoted as $F_1...F_5$. Thanks to these fixed regions, we ensure that

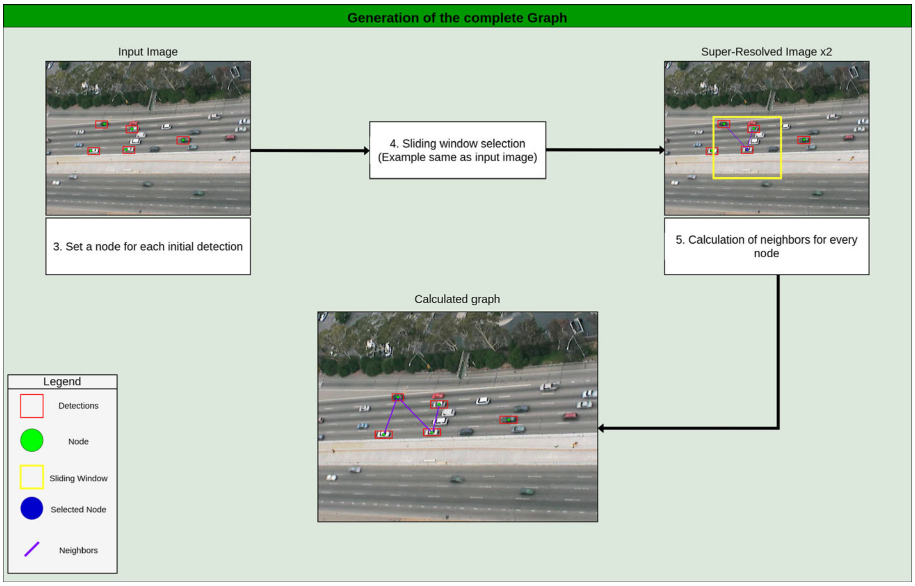
Generation of the complete Graph
Input Image
Super-Resolved Image x2
4. Sliding window selection (Example same as input image)
3. Set a node for each initial detection
5. Calculation of neighbors for every node
Calculated graph
Legend
Detections
Node
Sliding Window
Selected Node
Neighbors

Fig. 2 Generation of the complete graph flow

our proposal is applied over all regions of the image denoted as $\mathbf{X}_{\mathrm{LR}}$ in case the model does not obtain any initial detections.

Then, as shown in step 10 of Fig. 1, object detection is performed on the set of generated sub-images denoted as $\mathbf{I}'$, thus obtaining a list S, with the detections of each sub-image $\mathbf{I}'_k$ where N_i is the number of detections for sub-image $\mathbf{I}_k$.

$$S_i = \left\{ \left(\tilde{x}_{i,j}, \tilde{y}_{i,j}, \tilde{h}_{i,j}, \tilde{w}_{i,j}, \tilde{l}_{i,j}, \tilde{s}_{i,j} \right) \| j \in \{1, ..., N_i\} \right\} \tag{6}$$

In step 11, the object detections of S_i are computed in coordinates of $\tilde{\mathbf{X}}_{\mathrm{HR}}$. Therefore, they must be translated into coordinates of $\mathbf{X}_{\mathrm{LR}}$. The equation to convert a point $\tilde{\mathbf{h}}$ expressed in coordinates of $\mathbf{X}_i$ to coordinates $\mathbf{h}$ of $\mathbf{X}_{\mathrm{LR}}$ is as follows:

$$\mathbf{h} = \mathbf{y}_i + \frac{1}{Z}\tilde{\mathbf{h}} \tag{7}$$

where $\mathbf{y}_i$ is calculated as follows:

$$\mathbf{y}_i = \left(\frac{x_i + w_i}{2}, \frac{y_i + h_i}{2} \right) \tag{8}$$

$$\left(a_{i,j}, b_{i,j} \right) = \mathbf{y}_i + \frac{1}{Z}\left(\tilde{x}_{i,j}, \tilde{y}_{i,j} \right) \tag{9}$$

$$\left(c_{i,j}, d_{i,j} \right) = \mathbf{y}_i + \frac{1}{Z}\left(\tilde{h}_{i,j}, \tilde{w}_{i,j} \right) \tag{10}$$

$$q_{i,j} = \tilde{l}_{i,j} \tag{11}$$

$$r_{i,j} = \tilde{s}_{i,j} \tag{12}$$

Consequently, the set of object detections for the sub-image $\mathbf{I}'$ expressed in coordinates of $\mathbf{X}_{LR}$ is:

$$S'_i = \left\{ \left(a_{i,j}, b_{i,j}, c_{i,j}, d_{i,j}, q_{i,j}, r_{i,j} \right) \| j \in \{1, ..., N_i\} \right\} \tag{13}$$

where N_i is the number of detections for every sub-image generated, $(a_i, b_i) \in \mathbb{R}^2$ are the coordinates of the upper left corner of the ith detection within the sub-image $\mathbf{I}'_{\mathbf{k}}$, $(c_i, d_i) \in \mathbb{R}^2$ are the coordinates of the lower right corner of the ith detection, q_i is the class label of the detection, and $r_i \in \mathbb{R}$ is the class score of the detection.

Finally, in step 12, a cluster K is computed, grouping the translated detections coming from the optimal sub-images generated $\mathbf{I}'$ according to the Intersection over Union (IoU) measure computed on their associated bounding box for each pair of detections S'_j and S'_k.

$$\mathrm{IOU} = \frac{\mathrm{Area}\left(S'_j \cap S'_k \right)}{\mathrm{Area}\left(S'_j \cup S'_k \right)} \tag{14}$$

The clustering operation is performed for simultaneous group detections of the same element. The cluster K conforms to a list with the detections obtained for each element i. According to this list, the detection with the highest score for each element is selected. At the end of this process, an image with a higher number of detections and improved class inference of each element will be obtained. The official implementation is publicly available.[1].

4 Experiments

The objective is to determine the efficacy of our optimized proposal. For this purpose, sequences captured by video surveillance systems have been selected. A comparison of the following methods was made:

[1] https://github.com/IvanGarcia7/OSCOD.

- Original model (raw): the unmodified raw object detection model.
- SR Not Optimized: application of the proposal [31], based on applying a sub-image on which to re-infer for each element initially detected.
- Our proposal: an optimized framework. See Fig. 1 for more details.

The main difference between the methods referred to as SR Not Optimized [31] and our proposal is how sub-images for re-inference are generated. SR Not Optimized generates sub-images from the super-resolved image for each element initially identified in the low-resolution image. This approach has two limitations that our proposal addresses. Firstly, because it generates sub-images from initial detections, it may miss potential elements in certain areas and leave them unprocessed. In contrast, our proposal establishes five fixed regions to cover the entire image. Another aspect to consider is the processing time for a single frame. SR Not Optimized re-infers over each element, leading to a processing time proportional to the number of initial detections. On the other hand, our proposal includes an optimization module that minimizes the number of sub-frames to be processed based on the defined window size. Specific parameters used during the experiments are stated in Table 1.

Table 1 Selected values of the hyperparameters

Parameter	Value
Maximum number of detections per frame	300
Minimum percentage of inference	0.35
Window size R	0–100%

4.1 Pre-trained models

The proposal presented aims to improve the detection of small elements without modifying the convolutional neural network model or re-training it. Therefore, our proposal can be applied using any neural DCNN-based object detection model. Six pre-trained models have been selected from the TensorFlow Model Zoo repository:[2]

- CenterNet HourGlass104 Keypoints.
- CenterNet Resnet101 V1 FPN.
- Faster R-CNN Inception ResNet V2.
- EfficientDet D3.
- EfficientDet D4.
- EfficientDet D5.

These models have been trained with the COCO (Common Objects in Context) dataset [19], thus obtaining a generic model capable of detecting diverse classes in common areas, a widely used dataset as a reference in the literature.

4.2 Video sequences

Several sequences captured by video surveillance systems at high points have been selected. Four videos from *U.S. Highway 101 Dataset*[3] have been used. The systems are from the U.S. Department of Transportation and collect sequences of around 15 min each. It especially captures a large number of small- and medium-sized vehicles. These sequences have been manually annotated (632 manually labeled images with a total of 19343 vehicles) and used for quantitative and qualitative studies to evaluate our proposal. The results have been restricted to the category named *car* because the number of elements of that class for the four selected video sequences was the predominant one. We have also tested a series of frames from the dataset named *VisDrone (Vision Meets Drones)* [32].

4.3 Metrics

The average number of generated sub-images to re-infer has been used to evaluate the proposed methodology. This metric is directly variable according to the window size previously determined in the proposed methodology. It is crucial to optimize the generated images, decreasing the number of images on which to re-infer. This indicator makes it possible to determine the optimal window size that balances achieving high accuracy and reducing the processing time required for a particular sequence. A larger window size would allow for speeding up the processing time. However, it could result in a lower accuracy due to omitting some small elements. A smaller window size may improve accuracy but significantly increase the time required. This metric is essential for applications where processing time is a decisive factor or requires processing large datasets.

Another metric used is the COCO (Common Objects in Context) evaluator framework,[4] widely used in object detection and segmentation tasks. It is a standard reference for evaluating the models' performance. For this reason, the mean accuracy (mAP) has been selected as the evaluation metric, thus allowing an exhaustive, reliable, and objective evaluation of the quality of the detections provided by the model when applying the methodology presented. It computes the average precision value

[2] https://github.com/tensorflow/models/blob/master/research/object_detection/g3doc/tf2_detection_zoo.md.

[3] https://www.fhwa.dot.gov/publications/research/operations/07030/index.cfm.

[4] https://github.com/cocodataset/cocoapi/blob/master/PythonAPI/pycocotools/cocoeval.py.

Table 2 Average sub-images processed per frame obtained for sequence 1 of the U.S. Department of Transportation for the different pre-trained models used based on different window sizes R of our proposal

Sequence 1—NGSIM dataset sb-camera1-0820am-0835am						
	Models					
Proposal	CenterNet HourGlass104 Keypoints	CenterNet Resnet101 V1 FPN	Faster R-CNN Inception ResNet V2	EfficientDet D3	EfficientDet D4	EfficientDet D5
SR Not Optimized	24.503 ± 5.750	25.350 ± 4.374	22.175 ± 5.578	19.593 ± 4.443	29.316 ± 2.878	27.322 ± 3.429
OURS R = 0%	29.503 ± 5.750	30.350 ± 4.374	27.175 ± 5.578	24.593 ± 4.443	34.316 ± 2.878	32.322 ± 3.429
OURS R = 10%	28.582 ± 5.523	29.667 ± 4.086	25.119 ± 4.900	24.582 ± 4.430	33.836 ± 2.621	31.780 ± 3.110
OURS R = 20%	22.458 ± 4.457	23.780 ± 3.826	19.915 ± 3.803	20.124 ± 3.825	27.458 ± 2.772	25.525 ± 3.164
OURS R = 30%	21.045 ± 4.457	22.593 ± 3.861	18.107 ± 3.880	19.339 ± 3.792	26.316 ± 2.999	24.661 ± 2.846
OURS R = 40%	18.678 ± 3.811	20.232 ± 3.084	16.542 ± 3.469	17.565 ± 3.379	24.028 ± 2.530	22.531 ± 2.482
OURS R = 50%	16.435 ± 3.004	16.921 ± 2.291	14.938 ± 2.913	14.797 ± 2.401	20.503 ± 1.826	19.260 ± 1.899
OURS R = 60%	14.232 ± 2.154	14.282 ± 1.889	14.085 ± 2.405	12.808 ± 1.889	17.141 ± 1.809	15.870 ± 1.788
OURS R = 70%	12.960 ± 2.065	13.051 ± 1.668	13.226 ± 2.040	11.599 ± 1.698	15.768 ± 1.956	14.277 ± 1.800
OURS R = 80%	11.797 ± 2.037	12.169 ± 1.524	12.328 ± 1.628	10.797 ± 1.455	14.299 ± 1.858	12.972 ± 1.567
OURS R = 90%	10.610 ± 1.928	10.842 ± 1.243	11.006 ± 1.550	10.175 ± 1.257	12.299 ± 1.502	11.373 ± 1.224
OURS R = 100%	9.294 ± 1.232	10.056 ± 0.955	9.977 ± 1.266	9.831 ± 1.209	11.045 ± 1.413	10.497 ± 1.160

Table 3 Mean average precision obtained for the first sequence according to different window sizes R using the EfficientDet D4 model. A detection is considered valid when having confidence that exceeds the set threshold of 35%

Sequence 1—NGSIM dataset sb-camera1-0820am-0835am—EfficientDet D4—confidence > 0.35					
Proposal	IoU = 0.5:0.95-all	IoU > 0.5-all	IoU > 0.75-all	IoU = 0.5:0.95-small	IoU > 0.5-medium
Raw	0.145	0.188	0.187	0.137	0.218
SR Not Optimized [31]	0.47	0.697	0.588	0.468	0.484
Ours R = 0%	0.591	0.848	0.748	0.598	0.539
Ours R = 10%	0.591	0.848	0.747	0.598	0.541
Ours R = 20%	0.588	0.849	0.749	0.593	0.547
Ours R = 30%	0.584	0.838	0.75	0.591	0.525
Ours R = 40%	0.59	0.848	0.762	0.597	0.541
Ours R = 50%	0.586	0.838	0.749	0.593	0.518
Ours R = 60%	0.588	0.839	0.754	0.596	0.516
Ours R = 70%	0.585	0.839	0.752	0.593	0.512
Ours R = 80%	0.584	0.829	0.753	0.593	0.503
Ours R = 90%	0.572	0.819	0.736	0.583	0.473
Ours R = 100%	0.563	0.808	0.725	0.576	0.448

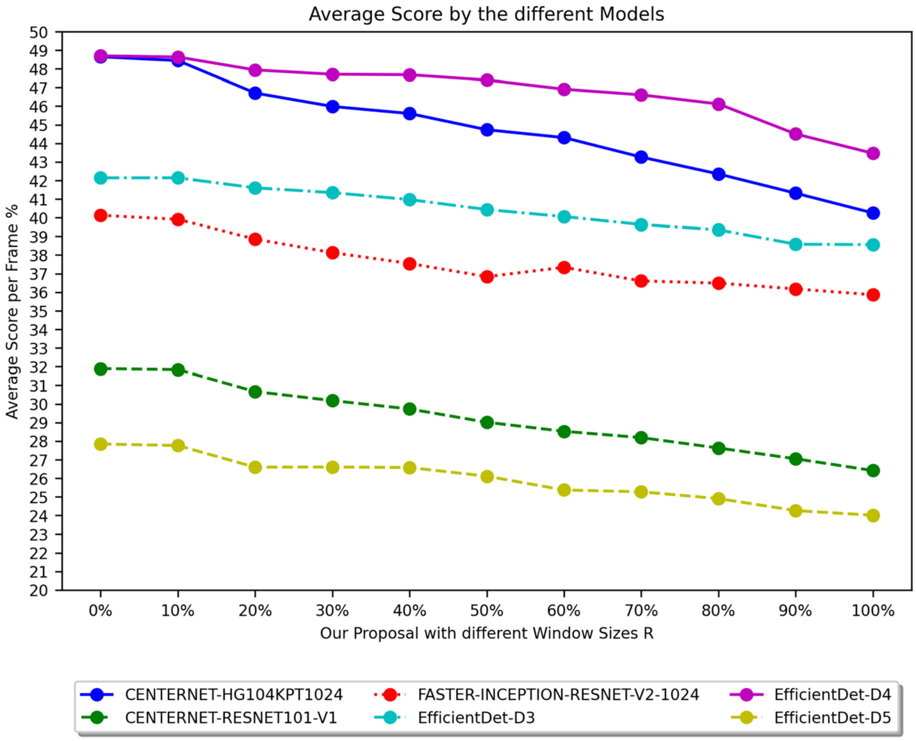

Fig. 3 Average score per frame using different windows sizes *R* for sequence 1 of the U.S. Department of Transportation

determining how accurate the model gives the predictions. The average precision is calculated over multiple IoU (Intersection over Union), the minimum area selected based on the annotation set as GT (ground truth), and the one obtained by the model to consider a coincidence as positive.

Finally, the average confidence (score) of the detections obtained by the model in a sequence provides information on the quality of the detections, allowing a comparison based on the selected window size to determine how it influences this value.

4.4 Results

Our optimized proposal requires the selection of a tunable parameter *R* to calculate the elements called neighbors. The center of the bounding box of the first object is calculated, and a defined window size *R* is used to create a region to determine whether two elements are neighbors. If the second element is entirely within this region, it is considered a neighbor. For each video sequence, it will be necessary to determine the optimal size of *R* to reduce the workload on which the model will have to re-infer. According to the set value, the number of sub-images to re-infer will decrease as *R* increases. Table 2 shows the number of sub-images generated according to the different window sizes set and their respective standard deviation. We can highlight how at first, *SR Not Optimized* processes a smaller number of sub-images than our proposal with a window size of 0%. This difference is due to including the five fixed regions in the image to process it completely, thus avoiding cases where no initial tentative areas are detected. As the window size increases, we can decrease the number of images required for processing by up to 60%, for example, in the CenterNet HourGlass104 Keypoints model compared to the SR Not Optimized method. It can be verified that as the window size is increased to a size greater than 10%, the number of optimal candidate regions is significantly minimized. The window size calculates the neighbors of two detections. Reducing the number of candidate areas with a relatively small window is difficult because the cliques will be conformed directly for only each initially detected element. Overall, an average speedup of 57.9% is obtained for the six models evaluated by the SR Not Optimized technique compared to the methodology presented with a window size of 100%.

Table 3 shows the mAP obtained for the non-optimized proposal denoted as *SR Not Optimized*, as well as for each of the applied window sizes of our proposal. First, we can determine that the mean average rate (mAP) obtained by the RAW model is low, obtaining an overall score of 0.145 and 0.137 for small elements. Relating the results of Tables 2 and 3, the overall mAP increases considerably using our optimized proposal, obtaining a 0.59 accuracy rate and 0.597 for small-sized elements, processing 16.67% fewer images with a window size *R* of 40% for the EfficientDet D4 model. We achieve higher accuracy than the one reached with the *SR Not Optimized* proposal,

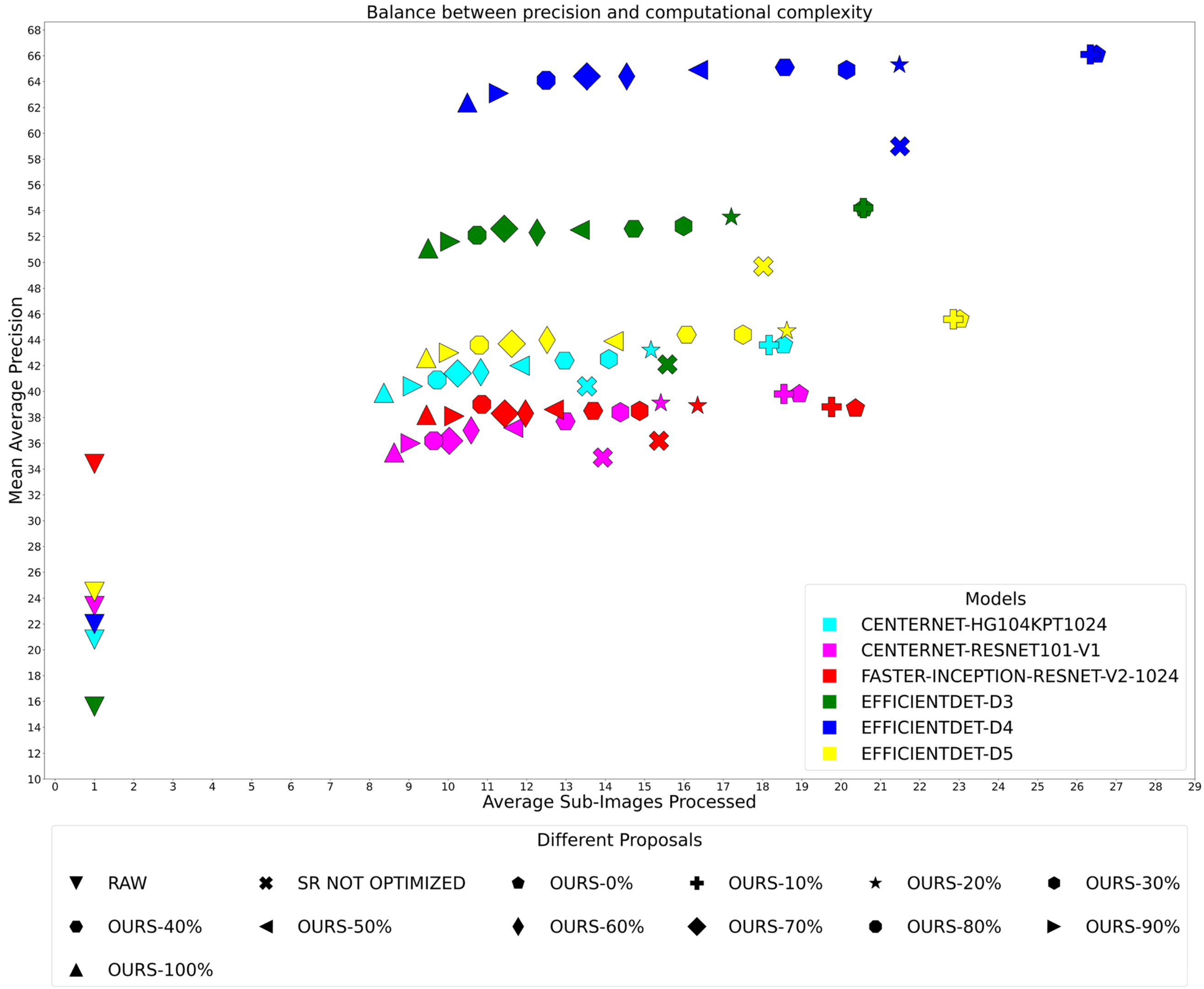

Fig. 4 Balance between precision and computational complexity. Average sub-images processed are presented on *x-axis*. *y-axis* represents the mean average precision (mAP). Both axis coordinates have been calculated using the mean of the four sequences of the U.S. Department of Transportation using two competitors and our proposal with different windows sizes *R*

computing fewer sub-images. This fact happens in the rest of the window sizes of our proposal, obtaining in the most restrictive case (R = 100%) a 0.563 accuracy compared with *SR Not Optimized* with 0.47.

Figure 3 shows the average confidence obtained for each element detected in sequence one by modifying the parameter R. As the size of the window increases, the number of sub-images to be processed is reduced, which implies that the same object could be detected fewer times. According to the results, the average score obtained for the detected elements is not significantly reduced according to the selected window size. Figure 4 represents the balance between precision and computational complexity. The x-axis of the figure sets the average number of sub-images required to re-infer by each proposal. At the same time, the y-axis represents the general mean average precision (mAP) for the mean of the result obtained for the four sequences. The *RAW* proposal obtains the worst results since, even though it only has to process the input image once, the mAP obtained is very low. We set the best solutions to proposals close to the upper left corner since this will mean that it has obtained the highest mAP by processing the least number of sub-images. We can therefore determine that our proposal significantly outperforms *SR Not Optimized*, getting a better mAP by reducing the computation in terms of sub-image processing. It is worth mentioning the good synergy that the application of our proposal has with models of the *EfficientDet* family since the results obtained are more than four times higher than the RAW model. Our proposal obtains the best results even in the most restrictive case with window size R = 100%. To better illustrate the increase in the detection rate

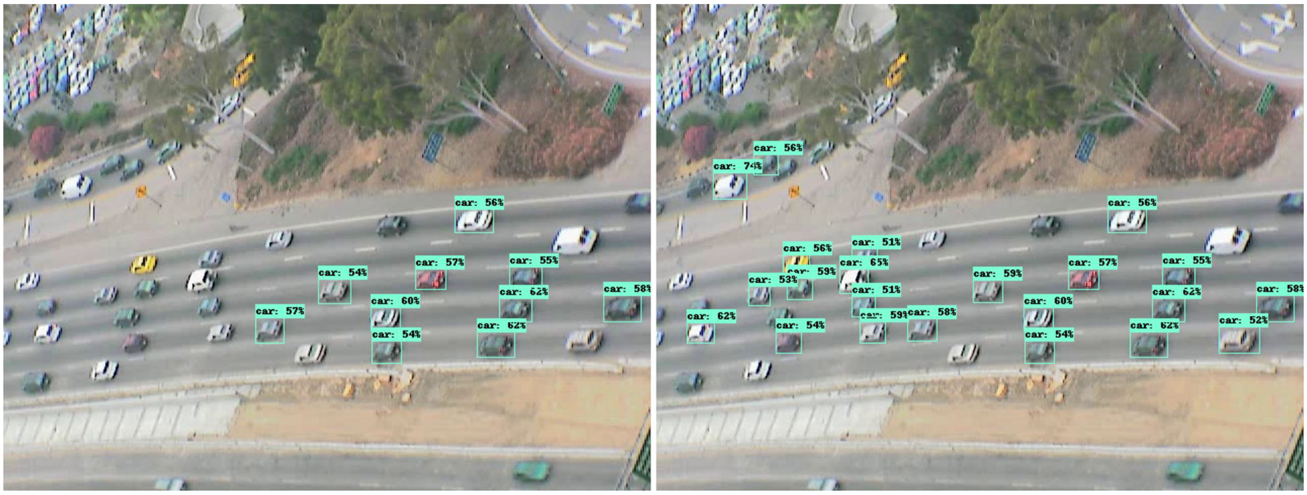

Fig. 5 Frame 3 of the first video denoted as sb-camera1-0820am-0835am processed by CenterNet HourGlass 104 Kpts with a confidence > 50%. The left side shows the results obtained by the raw model, while the right side shows the detections after applying our optimized proposal with $R = 50\%$

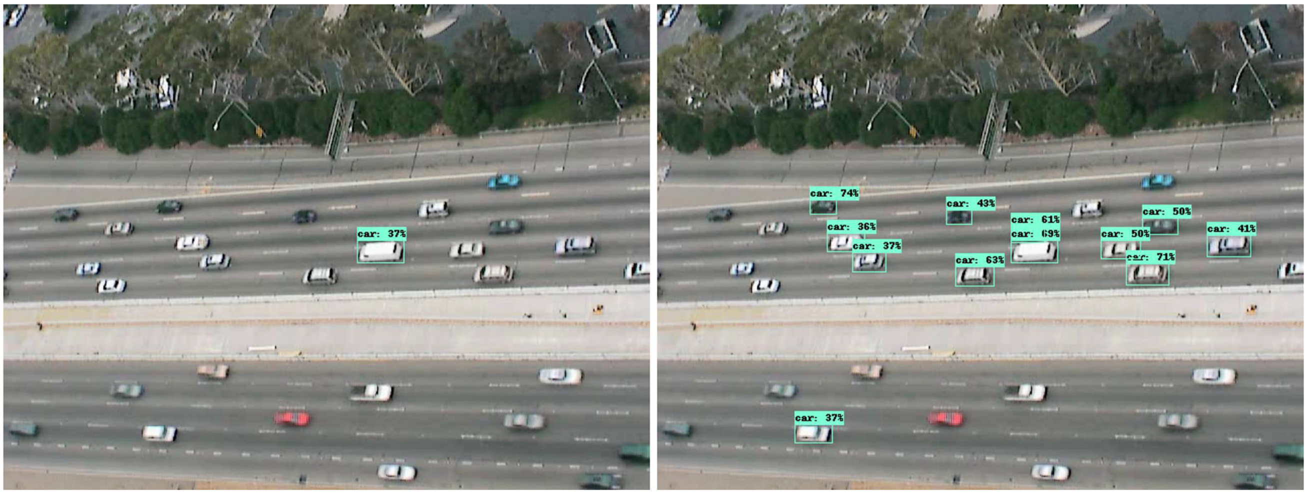

Fig. 6 Frame 134 of the second video denoted as sb-camera3-0750am-0805am processed by EfficientDet D3 with a confidence > 35%. The left side shows the results obtained by the raw model, while the right side shows the detections after applying our optimized proposal with $R = 50\%$

of our proposal compared to the original model, a series of qualitative results are shown, represented in Figs. 5, 6, and 7. To demonstrate the present improvement through the use of this proposal, we have also tested a series of frames from the dataset named *VisDrone (Vision Meets Drones)* [32], see Figs. 8 and 9. According to the results discussed in this section, implementing the optimized proposal improves the accuracy of the elements initially detected by the model but also detects objects not identified a priori.

5 Conclusions

This paper proposes an optimization algorithm to improve the technique's speed presented previously in [31]. First, the detector model sets tentative elements to perform super-resolution. After that, it generates super-resolved sub-images to re-infer on it. The proposed greedy algorithm maximizes the cliques which contain the largest possible number of elements, minimizing the number of sub-images on which to infer. These results in more

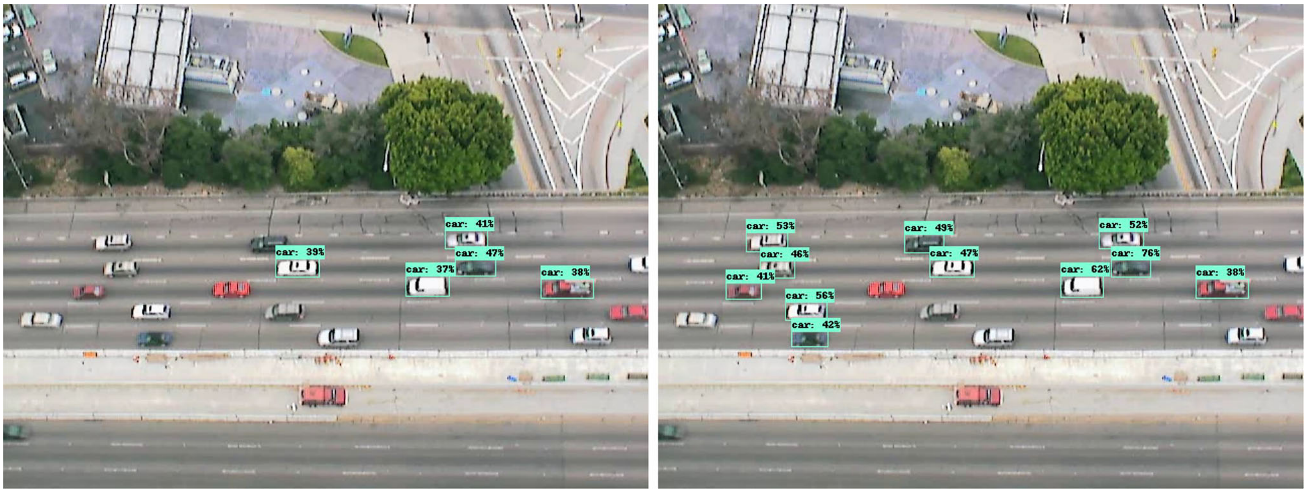

Fig. 7 Frame 32 of the third video denoted as sb-camera4-0820am-0835am processed by EfficientDet D4 with a confidence > 35%. The left side shows the results obtained by the raw model, while the right side shows the detections after applying our optimized proposal with $R = 50\%$

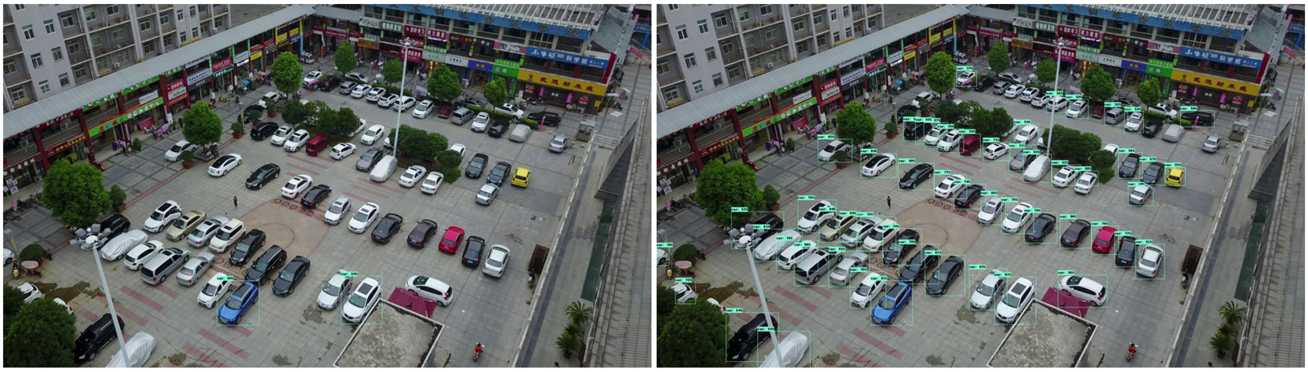

Fig. 8 Frame 0000001_05499_d_0000010 of the VisDrone dataset [32] processed by EfficientDet D4 with a confidence > 50%. The left side shows the results obtained by the raw model, while the right side shows the detections after applying our optimized proposal with $R = 50\%$

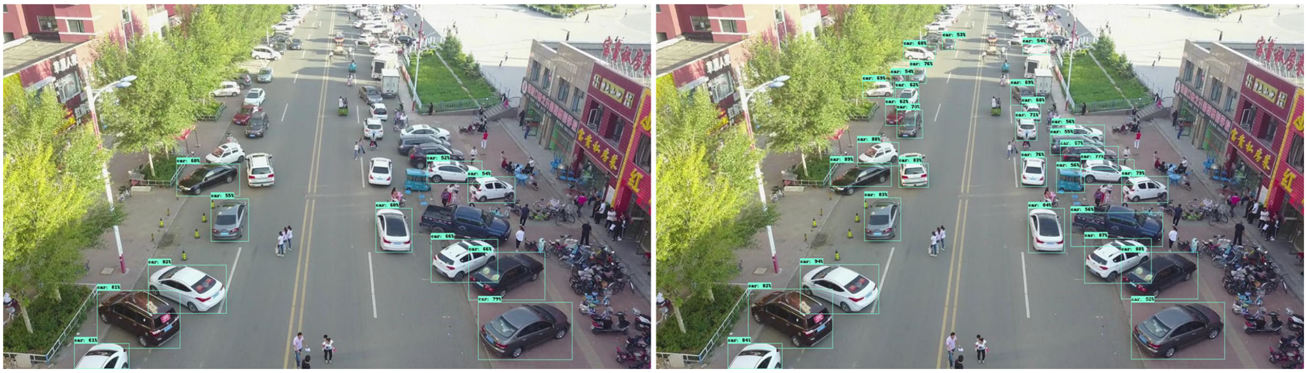

Fig. 9 Frame 0000213_03920_d_0000243 of the VisDrone dataset [32] processed by EfficientDet D4 with a confidence > 35%. The left side shows the results obtained by the raw model, while the right side shows the detections after applying our optimized proposal with $R = 50\%$

accurate estimates of the characteristics of previously detected objects and the detection of additional objects that the object detection network did not initially detect. The advantage of our proposal is that it avoids modifying the internal structure of the model, as well as re-training for a specific scene. The results determine that the application of our proposal improves the mean average precision (mAP) compared with the raw model or the direct application of super-resolution to the full image given as input. We can highlight *EfficientDet D4* whose *mAP* is considerably increased, going from an accuracy of 14.5% obtained by the pre-trained model up to 59.1% when applying the optimized approach. Furthermore, experimental results show that increasing the window size can lead to better results, thereby reducing the processing time required for some cases. However, it is essential to perform a preliminary analysis based on the area of application to determine the optimal parameters of the proposed approach. This analysis can ensure the methodology is optimally applied to provide the best balance between accuracy and time required. Overall, our proposed approach provides a flexible and efficient solution to improve the accuracy of convolutional neural models to optimize specific application areas.

According to future lines, there are several avenues to pursue for the future development of the proposed methodology. Firstly, incorporating techniques aimed at feature selection is contemplated to improve the performance on the areas of interest in a sequence where the camera is placed in a static position. Some of these methods could be SIFT or SURF, among others. Applying these methods would allow the selection of relevant regions, improving the localization of objects, thus improving their accuracy after re-inferring with the convolutional neural network. Additionally, integrating temporal information for object detection in static video sequences could be another promising direction, as it is often required to track objects captured by a static video surveillance system.

Acknowledgements This work was partially supported by the Autonomous Government of Andalusia (Spain) under project UMA20-FEDERJA-108, project name Detection, characterization and prognosis value of the non-obstructive coronary disease with deep learning. It includes funds from the European Regional Development Fund (ERDF). It was also partially supported by the University of Málaga (Spain) under grants B1-2019_01, project name Anomaly detection on roads by moving cameras; B1-2019_02, project name Self-Organizing Neural Systems for Non-Stationary Environments; B1-2021_20, project name Detection of coronary stenosis using deep learning applied to coronary angiography, B4-2022, project name Intelligent Clinical Decision Support System for Non-Obstructive Coronary Artery Disease in Coronarographies, and B1-2022_14, project name Deteccion de trayectorias anomalas de vehiculos en camaras de trafico. Iván García-Aguilar was funded by a scholarship from the Autonomous Government of Andalusia (Spain) under the Young Employment operative program [grant number SNGJ5Y6-15]. The authors thankfully acknowledge the computer resources, technical expertise, and assistance provided by the SCBI (Supercomputing and Bioinformatics Center) of the University of Málaga. They also gratefully acknowledge the support of NVIDIA Corporation with the donation of an RTX A6000 GPU with 48Gb. The authors also thankfully acknowledge the grant of the Universidad de Málaga and the Instituto de Investigación Biomédica de Málaga y Plataforma en Nanomedicina-IBIMA Plataforma BIONAND.

Funding Funding for open access publishing: Universidad Málaga/CBUA.

Data availability Sequences analyzed during the current study are available on the official Web site (https://www.fhwa.dot.gov/publications/research/operations/07030/index.cfm). Labeled annotations are available in the GitHub repository (https://github.com/IvanGarcia7/NGSIM-Dataset-Annotations).

Declarations

Conflict of interest The authors declare that they have no known competing financial interest or personal relationships that could have appeared to influence the work reported in this paper.

References

1. Shen L, Tao H, Ni Y, Wang Y, Stojanovic V (2023) Improved Yolov3 model with feature map cropping for multi-scale road object detection. Meas Sci Technol 34(4):045406. https://doi.org/10.1088/1361-6501/acb075
2. Zhuang Z, Tao H, Chen Y, Stojanovic V, Paszke W (2023) An optimal iterative learning control approach for linear systems with nonuniform trial lengths under input constraints. IEEE Trans Syst Man Cybern Syst 53(6):3461–3473. https://doi.org/10.1109/TSMC.2022.3225381
3. Zhou L, Tao H, Paszke W, Stojanovic V, Yang H (2020) Pd-type iterative learning control for uncertain spatially interconnected systems. Mathematics 8(9):1528. https://doi.org/10.3390/math8091528
4. Redmon J, Divvala S, Girshick R, Farhadi A (2016) You only look once: unified, real-time object detection. In: IEEE conference on computer vision and pattern recognition (CVPR), pp 779–788 https://doi.org/10.1109/CVPR.2016.91
5. Liu W, Anguelov D, Erhan D, Szegedy C, Reed S, Fu C-Y, Berg AC (2016) Ssd: single shot multibox detector. Lecture Notes in Computer Science, Springer International Publishing, pp 21–37 https://doi.org/10.1007/978-3-319-46448-0_2.
6. Lee Y, Hwang J-w, Lee S, Bae Y, Park J (2019) An energy and GPU-computation efficient backbone network for real-time object detection. https://arxiv.org/abs/1904.09730
7. Benito-Picazo J, Domínguez E, Palomo E, López-Rubio E (2020) Deep learning-based video surveillance system managed by low cost hardware and panoramic cameras. Integr Comput-Aided Eng 27:1–15. https://doi.org/10.3233/ICA-200632
8. Ren S, He K, Girshick R, Sun J (2016) Faster R-CNN: towards real-time object detection with region proposal networks
9. Tan M, Le Q (2019) EfficientNet: rethinking model scaling for convolutional neural networks. In: Proceedings of the 36th international conference on machine learning, vol 97, pp 6105–6114

10. Sandler M, Howard A, Zhu M, Zhmoginov A, Chen L-C (2019) MobileNetV2: inverted residuals and linear bottlenecks
11. Liu W, Anguelov D, Erhan D, Szegedy C, Reed S, Fu C-Y, Berg AC (2016) SSD: Single shot MultiBox detector. In: Computer vision–ECCV 2016, Springer International Publishing, pp 21–37 https://doi.org/10.1007/978-3-319-46448-0_2
12. Bochkovskiy A, Wang C-Y, Liao H-YM (2020) YOLOv4: optimal speed and accuracy of object detection
13. Zhu X, Lyu S, Wang X, Zhao Q (2021) TPH-YOLOv5: improved YOLOv5 based on transformer prediction head for object detection on drone-captured scenarios
14. Subudhi BN, Nanda PK, Ghosh A (2011) A change information based fast algorithm for video object detection and tracking. IEEE Trans Circuits Syst Video Technol 21(7):993–1004. https://doi.org/10.1109/TCSVT.2011.2133870
15. Mandel T, Jimenez M, Risley E, Nammoto T, Williams R, Panoff M, Ballesteros M, Suarez B (2023) Detection confidence driven multi-object tracking to recover reliable tracks from unreliable detections. Pattern Recogn 135:109107. https://doi.org/10.1016/j.patcog.2022.109107
16. Kavitha R, Chitra D (2021) An extreme learning machine and action recognition algorithm for generalized maximum clique problem in video event recognition. Dyn Syst Appl 30(8):1228–1249
17. Ren S, Li J, Tu T, Peng Y, Jiang J (2021) Towards efficient video detection object super-resolution with deep fusion network for public safety. Secur Commun Netw 2021:1–14. https://doi.org/10.1155/2021/9999398
18. Deng J, Dong W, Socher R, Li L-J, Li K, Fei-Fei L (2009) Imagenet: a large-scale hierarchical image database. In: 2009 IEEE conference on computer vision and pattern recognition, IEEE, pp 248–255
19. Lin T-Y, Maire M, Belongie S, Bourdev L, Girshick R, Hays J, Perona P, Ramanan D, Zitnick CL, Dollár P (2015) Microsoft COCO: common objects in context
20. Tan M, Pang R, Le QV (2020) EfficientDet: scalable and efficient object detection
21. Rabbi J, Ray N, Schubert M, Chowdhury S, Chao D (2020) Small-object detection in remote sensing images with end-to-end edge-enhanced gan and object detector network. Remote Sens 12(9):1432. https://doi.org/10.3390/rs12091432
22. Deng C, Wang M, Liu L, Liu Y, Jiang Y (2022) Extended feature pyramid network for small object detection. IEEE Trans Multimed 24:1968–1979. https://doi.org/10.1109/TMM.2021.3074273
23. Su P, Li W, Sha L, Shi Y, Dong T (2021) Traffic sign recognition algorithm based on feature pyramid attention. In: Journal of physics: conference series, vol 2035, Chapter 1
24. Khan K, Imran A, Rehman HZU, Fazil A, Zakwan M, Mahmood Z (2021) Performance enhancement method for multiple license plate recognition in challenging environments. Eurasip J Image Video Process 2021(1):1–23
25. Dong C, Loy CC, He K, Tang X (2014) Learning a deep convolutional network for image super-resolution. In: Fleet D, Pajdla T, Schiele B, Tuytelaars T (eds) Computer vision-ECCV 2014. Springer International Publishing, Cham, pp 184–199
26. Dong C, Loy CC, Tang X (2016) Accelerating the super-resolution convolutional neural network. In: CoRR arXiv:1608.00367
27. Kim J, Lee JK, Lee KM (2015) Accurate image super-resolution using very deep convolutional networks. In: CoRR arXiv:1511.04587
28. Kim J, Lee JK, Lee KM (2015) Deeply-recursive convolutional network for image super-resolution. In: CoRR arXiv:1511.04491
29. Kong D, Han M, Xu W, Tao H, Gong Y (2006) Video super-resolution with scene-specific priors. In: Procedings of the British machine vision conference 2006, British Machine Vision Association https://doi.org/10.5244/c.20.57
30. Camargo A, He Q, Palaniappan K (2012) Performance evaluation of optimization methods for super-resolution mosaicking on UAS surveillance videos. In: Holst GC, Krapels KA (eds) SPIE proceedings. SPIE
31. García-Aguilar I, Luque-Baena RM, López-Rubio E (2021) Improved detection of small objects in road network sequences using CNN and super resolution. Expert Syst 39(2):e12930. https://doi.org/10.1111/exsy.12930
32. Zhu P, Wen L, Du D, Bian X, Fan H, Hu Q, Ling H (2021) Detection and tracking meet drones challenge. IEEE Trans Pattern Anal Mach Intell 44:7380–7399. https://doi.org/10.1109/TPAMI.2021.3119563