A Lightweight Gaussian-Based Model for Fast Detection and Classification of Moving Objects

Joaquin Palma-Ugarte[1], Laura Estacio-Cerquin[2,3], Victor Flores-Benites[4] and Rensso Mora-Colque[1]
[1]*Department of Computer Science, Universidad Católica San Pablo, Arequipa, Peru*
[2]*Department of Radiology, The Netherlands Cancer Institute - Antoni van Leeuwenhoek Hospital, Amsterdam, The Netherlands*
[3]*GROW School for Oncology and Developmental Biology, Maastricht University, Maastricht, The Netherlands*
[4]*Universidad de Ingeniería y Tecnología – UTEC, Lima, Peru*

Keywords: Detection, Classification, Moving Objects, Gaussian Mixture, Lightweight Model.

Abstract: Moving object detection and classification are fundamental tasks in computer vision. However, current solutions detect all objects, and then another algorithm is used to determine which objects are in motion. Furthermore, diverse solutions employ complex networks that require a lot of computational resources, unlike lightweight solutions that could lead to widespread use. We introduce TRG-Net, a unified model that can be executed on computationally limited devices to detect and classify just moving objects. This proposal is based on the Faster R-CNN architecture, MobileNetV3 as a feature extractor, and a Gaussian mixture model for a fast search of regions of interest based on motion. TRG-Net reduces the inference time by unifying moving object detection and image classification tasks, and by limiting the regions of interest to the number of moving objects. Experiments over surveillance videos and the Kitti dataset for 2D object detection show that our approach improves the inference time of Faster R-CNN (0.221 to 0.138s) using fewer parameters (18.91 M to 18.30 M) while maintaining average precision (AP=0.423). Therefore, TRG-Net achieves a balance between precision and speed, and could be applied in various real-world scenarios.

1 INTRODUCTION

The detection and classification of moving objects are fundamental tasks in many day-to-day systems, from smart surveillance to autonomous driving and activity recognition. They all have in common the processing of images and videos in order to decode their content and obtain useful information in real time. Today, videos constitute an extensive source of information that is rarely analyzed. Despite the progressive and continuous growth of computer vision solutions based on deep learning models, the task of detecting and classifying moving objects in video is still a challenging and attractive area with a lot of active research and development in the industry.

The rise of CNNs has been advantageous for the development of complex and precise systems. Currently, many state-of-the-art models and frameworks are capable of accurately detecting objects in real time. Focusing mainly on the detection of humans (Gruosso et al., 2021), vehicles (Chen and Hu, 2021), and even pets (Yuan, 2021). However, there are a limited number of studies on moving object detection, either by applying supervised learning through CNNs or other unsupervised methods with less computational demand.

Moving object detection (MOD) is the detection of moving objects with respect to the surrounding area or region of a sequence of video frames. For example, distinguish moving pedestrians from static buildings in a video recorded by a surveillance camera. Static objects are called 'background', and moving objects are called 'foreground'. This task constitutes the basic step for various subsequent specific tasks, such as the classification or tracking of moving objects. Generally, advanced video analysis consists of three main phases: (1) the identification of the moving target, (2) the tracking of the identified object in a given series of video frames, and (3) the analysis of the object's movement to determine its behavior. Therefore, the detection of moving objects is a fundamental task for most complex video analysis processes (Kulchandani and Dangarwala, 2015).

On the other hand, image classification refers to

Palma-Ugarte, J., Estacio-Cerquin, L., Flores-Benites, V. and Mora-Colque, R.
A Lightweight Gaussian-Based Model for Fast Detection and Classification of Moving Objects.
DOI: 10.5220/0011697200003417
In *Proceedings of the 18th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP 2023) - Volume 5: VISAPP*, pages 173-184
ISBN: 978-989-758-634-7; ISSN: 2184-4321

the task of analyzing an image and identifying the class to which it belongs. In essence, a class is a label i.e. 'truck', 'person', 'cat', etc. This task is a common next step to moving object detection; once moving objects are detected, it is reasonable to discover what has moved. Although motion detection and classification are two different issues, treating them separately generates solutions either very specific or very generic. This research attempts to build a unified approach, as current solutions do not consider the movement of objects as a critical criterion for object detection and classification.

An important aspect to consider when developing a computer vision solution is the size and execution time of the proposal. This heavily depends on the final device using the model. In our experience, the most popular models do not rely on high computational power, as they are commonly executed by computationally limited devices. Such devices are single-board computers with few resources, formalizing the definition, a computationally limited devices is a computer with no more than 4GB of memory and a processor with no more than 4 cores and 1.5GHz of speed (Glegoła et al., 2021). Although there are complex systems and architectures that delegate the work of detection and classification to servers in the cloud, such as (Alsmirat et al., 2017), these are dependent on the Internet connection and the passing of information through unsecured networks. This is inconvenient when dealing with sensitive systems that require privacy, security, and full availability in adverse situations (Zhang, 2021). Furthermore, they require expensive architectures to ensure data protection, scalability, and privacy; operations that can only be carried out by large companies (Haouari et al., 2018). There are new proposals that try to solve this problem by applying fog computing; however, the need for processing on local nodes to execute critical tasks remains a predominant property (Haouari et al., 2018). Consequently, we will focus on devices with limited computational power, which implies elaborating a lightweight and low latency model. Therefore, promoting the flexibility of the solution and its widespread use in different contexts.

The main contributions of this work are summarized as follows: (1) A novel approach that jointly addresses the well-known tasks of moving object detection and image classification. These tasks are usually performed separately, thus our proposal introduces a flexible model that unifies these tasks into one end-to-end architecture. (2) In the literature there are famous generic object detection models (Redmon et al., 2016; Liu et al., 2016). However, these models recognize all objects, even if they are moving or not; thus, another algorithm is necessary to determine which objects are in motion. In our approach, we propose a network that uses fewer computational resources to efficiently recognize and classify only moving objects. (3) Our model is inspired by Faster R-CNN (Ren et al., 2015) architecture, which allows us to employ a region proposal method based on Gaussian Mixtures (Zivkovic, 2004) instead of a Region Proposal Network (RPN). This grants an execution speed-up, and also the identification of movement.

2 RELATED WORK

We are addressing the detection and classification of moving objects using video data. This involves locating only moving objects, labeling them with a bounding box, and assigning them their respective class. To our knowledge, this task has not been previously tackled in a unified way. While video object detection is a similar problem, it does not necessarily imply attention to motion detection. Therefore, we have to review four concepts: (1) moving object detection, (2) image classification, (3) generic object detection, and (4) approaches based on movement detection.

2.1 Moving Object Detection

Moving object detection (MOD), also known as change detection, is the detection of non-stationary objects with respect to the surrounding area or region from a sequence of video frames (Kulchandani and Dangarwala, 2015). Note that the output of MOD methods is not a list of classes and bounding boxes; it is a binary mask of the image that highlights changing pixels. Current MOD methods can be broadly classified into (1) traditional methods and (2) deep learning methods. Although there are proposals that combine both approaches, they are often applied in specific contexts to solve real-world problems.

Traditional unsupervised methods do not require labeled data. They usually have two components: (1) a background model that initializes the background scene and updates it over time, and (2) a classifier that classifies each pixel as foreground or background (Hou et al., 2021). The classifier is generally a mathematical algorithm based on the output of the background model. On the other hand, there are many background modeling schemes. Early approaches used temporal and adaptive filters, including moving average filters (Yi and Liangzhong, 2010), temporal median filters (Hung et al., 2014), and the Kalman filter (Patel and Thakore, 2013). The next approaches used statistical and probabilistic representa-

tions to model the background, such as Gaussian mixtures models (Song et al., 2020) and semantic models (Braham et al., 2017). These probabilistic methods have shown better results than the sole use of filters. Novel approaches attempt to combine the use of temporal filters with probabilistic methods, formulating robust solutions for common real-world problems such as vehicle speed estimation (Tayeb et al., 2021). Since MOD does not represent the whole problem, it is not necessary to delve into complex frameworks, but it is advisable to pay attention to the benefits of probabilistic methods.

In general, the traditional methods are computationally fast and intuitive, showing high accuracy in static scenarios with little interference (Kulchandani and Dangarwala, 2015). However, they do not allow accurate detection in complex scenarios, as videos with shadows, dynamic backgrounds, and lighting changes are difficult to deal with. Consequently, deep learning methods have been a trend in addressing the aforementioned challenges.

The most representative models of deep learning are CNNs, which perform well on machine learning problems, especially in computer vision applications. However, they have also been applied to other fields, such as natural language processing (Albawi et al., 2017) with surprising results. Therefore, a manifest approach is to use a CNN instead of the traditional pixel classifiers. In (Braham and Van Droogenbroeck, 2016) the input is a single grayscale image, and the output is the probability of each pixel belonging to the foreground. If a probability exceeds a decision threshold, the pixel is considered foreground, otherwise background, forming a binary mask of the image. This straightforward approach inspired the development of more complex architectures, such as the use of hybrid spiking neural networks (Machado et al., 2021). Although this approach has many advantages, these models inherit the main problems of any solution based on deep learning: (1) obtaining quality data and (2) ensuring accuracy when unseen data are tested. Different and novel methods have been proposed to address these problems, such as the application of unsupervised learning (Yang et al., 2019), and self-supervised learning (Yang et al., 2022). However, the common denominator remains the increasing computational and data demand for testing and training.

2.2 Image Classification

Image classification has been fueled by the rise of CNNs since the introduction of Alex-Net (Krizhevsky et al., 2012) in 2012. CNNs are the most representative models of deep learning, as they can exploit the basic properties that underlie natural signals: (1) translation invariance, (2) local connectivity, and (3) composition hierarchies (Liu et al., 2020). This is highly relevant since a CNN-based solution addresses object classification through feature extraction.

There are many well-known architectures (Simonyan and Zisserman, 2014; He et al., 2016; Howard et al., 2017), yet elaborated architectures and improvements are constantly being developed (Xie et al., 2017; Tan and Le, 2019). Although CNNs were born as image classifiers, their use has become widespread as encoders and feature extractors. Working as backbone models of complex applications and frameworks (Zaidi et al., 2022).

2.3 Generic Object Detection

A generalization of image classification is object detection, the task of determining where objects are located and which category each object belongs to. Generic object detection models consist of three stages: (1) informative region selection, (2) feature extraction, and (3) classification (Zhao et al., 2019). CNN-based object detection architectures can be divided into two main groups: (1) two-stage detectors, and (2) one-stage detectors. The two-stage detectors generate region proposals first and then classify each proposal into their respective categories. While a one-stage detector considers object detection as a regression problem, achieving final results (categories and locations) directly (Liu et al., 2020).

In both cases, a CNN-based backbone is used to perform feature extraction. Although one-stage detectors have proven to be fast, they struggle with accuracy and flexibility when adapting their architecture to new tasks. Unlike two-stage detectors; that are slower, but accurate and flexible since they allow custom methods to determine regions of interest according to the addressed problem. The most widely used two-stage architecture is Faster R-CNN (Ren et al., 2015). Regarding the one-stage detectors, YOLO (Redmon et al., 2016) and SSD (Liu et al., 2016) stand out. Due to the nature of the proposal, it is reasonable to adapt a two-stage architecture to define regions based on the movement of objects.

2.4 Approaches Based on Movement Detection

In (Hou et al., 2021), it is proposed a lightweight three-dimensional CNN for moving object detection. This model is characterized by accepting multiple inputs and multiple outputs, and by using separable 3D

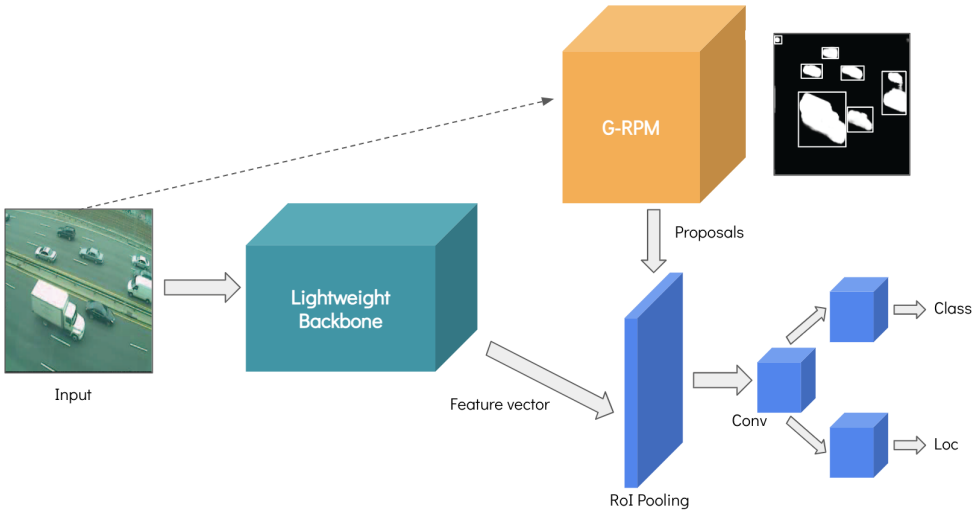

Figure 1: TRG-Net architecture. During inference, the input image passes through a backbone to obtain its feature vector. Then, the same input trains the G-RPM model that returns a list of region proposals. The previous two outputs pass through a pooling layer and fully connected layers that returns the classes and locations of the moving objects.

convolutions to explore spatiotemporal information in video data. This model addresses weight and latency issues using the concept of separable layers proposed by Mobile-Net (Howard et al., 2017), making the network an optimized detector for devices with limited memory and computational power. However, it does not include localization and classification as part of the pipeline. Furthermore, since it is a completely supervised solution, the authors highlight the limitations when dealing with unseen data. Here we can stand out that traditional motion detection techniques do not deal with the problem of data variety and over-fitting.

In (Jagannathan et al., 2021), it is proposed the construction of a traffic monitoring system that detects and classifies moving vehicles. This proposal applies a Gaussian mixture model to enhance the input images by detecting moving objects. These moving objects are cropped and sent to an ensemble of image classifiers, which outputs the final classification by majority voting. This approach can be very inefficient since various classifiers are executed for each detected object. Furthermore, it is similar to the first R-CNN architecture (Girshick et al., 2014), where the regions of interest are calculated previously and then passed one by one through an image classifier. Even though the problem is aligned with ours, the authors created a complex framework, but not a unified model, that performs both tasks. Moreover, it does not consider the lightness of the solution, resulting in an infeasible proposal for computationally limited devices.

In (Fan et al., 2021), it is proposed an optical-flow-based framework for video object detection. The authors identify occlusion as the main problem, arguing that it leads to appearance deterioration. However, the application of optical flow indirectly favors the detection of moving objects, since it enhances the foreground pixels. This framework follows this pipeline: (1) the video frames are grouped sharing the same optical flow feature map, (2) an enhanced image is formed by merging the shared feature map with the current video frame, and finally (3) an image is passed through an object detection model to get the object labels and bounding boxes. The results show that effective masking of background information can make the object detection model more focused on foreground objects. In contrast to this method, our proposal uses a change detection model to determine the regions of interest within the object detection model. This avoids the preprocessing time of the optical flow, reducing the complexity of our solution.

3 PROPOSED MODEL

To address the detection and classification of moving objects, we propose The Real Gaussian Network (TRG-Net). A solution based on the Faster R-CNN architecture (Ren et al., 2015), a lightweight backbone for feature extraction, and the use of a Gaussian Mixture Model (GMM) as the basis for searching regions of interest that contain moving objects. TRG-Net is a new model based on existing solutions that solve the specific problem of detecting and classifying moving objects in videos. Efficiency of the model is prioritized, thus, the design is oriented to the execution of the proposal in computationally limited de-

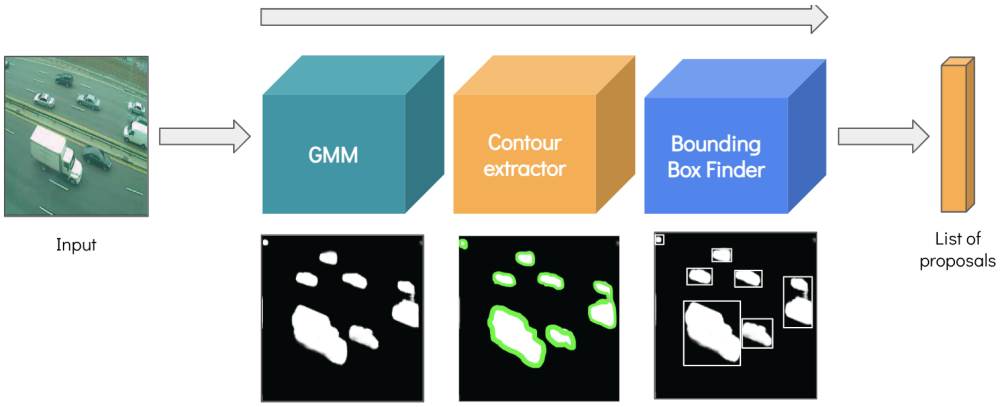

Figure 2: Gaussian-based Region Proposal Model.

vices.

This approach adopts the unified two-stage architecture for object detection proposed by region-based models, allowing us to use a novel region proposal model while maintaining the detection accuracy. Thus, a Gaussian-based Region Proposal Model (G-RPM) can be attached to the Faster R-CNN architecture to detect potential moving objects in consecutive video frames. The GMM does not seek to perform an accurate segmentation, but with high recall so as not to lose potential regions of interest. A detailed explanation of this region proposal model can be found in the following subsection 3.1.

The base version of TRG-Net employs the large version of MobileNetV3 (Howard et al., 2019) as the backbone. And, instead of the Region Proposal Network (RPN), it uses a G-RPM based on the GMM proposed by (Zivkovic, 2004). This architecture and the inference pipeline can be seen in Figure 1. The code is publicly available at: https://github.com/rodp63/TRG-net.

3.1 G-RPM

We introduce the Gaussian-based Region Proposal Model (G-RPM), a GMM-based model to find regions of interest. The input is an image, and the output is a list of bounding boxes containing potential moving objects. The model performs three steps. First, the GMM makes the classification of each pixel in the image as background or foreground, creating a binary mask over the input. Second, this mask is processed by a contour extractor (Suzuki et al., 1985), which returns a list of the object contours within the binary image. Finally, the bounding boxes of every object are calculated by finding the vertices with minimum and maximum values in the X and Y dimensions. This model can be seen in Figure 2.

Since there can be noise in the image due to lighting or weather conditions, many contours delimit an area close to zero. These contours are discarded thanks to the definition of a threshold a that determines the smallest area required to consider a foreground region as a potential moving object. This parameter depends on average size of the foreground objects in an input video, the base model of TRG-Net uses a threshold $a = 35px^2$. This value was calculated experimentally considering the size of the objects in the training data set, such as pedestrians or vehicles captured by a fixed camera. However, the threshold a can be dynamically modified during inference. Two important aspects need to be considered when setting this value: (1) a low value could identify noise as regions of interest, increasing the number of proposals and slowing down the inference time; on the other hand, (2) a large value could cause information loss by skipping small objects.

There is a second parameter in G-RPM, the learning rate lr of the GMM. This parameter is needed every time a new image updates the model, but it is defined once at construction time. However, there is an option for the algorithm to use an automatically chosen learning rate for every update. The value of lr is a floating number between 0 and 1, and indicates how quickly the GMM is learned. 0 means that the GMM is not updated at all and 1 means that the GMM is completely re-initialized from the last frame. It is advisable to try various lr values to determine which one best suits the input video images. You can start with the automatic value, and try from 0 to 1. As a general rule, values close to 0 are recommended since it indicates that the input has little noise and the proposals will be more precise.

To improve the recall of the model, a simple heuristic is applied in the Bounding Box Finder stage. Once the bounding box is calculated, it is contracted and expanded by a few units to include a larger number of proposals. These small deformations imply

more opportunities to correctly classify moving objects. This is relevant since a traditional change detector, such as GMM, is characterized by being fast, but not very accurate. Especially when the input has dynamic backgrounds or environmental phenomena.

3.2 Other Optimizations

To ensure that our proposal works with small tensors, the input image goes through a transformation pipeline that reduces its size if the input is too large. However, if the image is too small, this re-scaling helps to improve its resolution and thus the detection results. The base TRG-Net configuration scales the input tensors so that the image height is between 320 and 640 without losing the aspect ratio. The region proposals are also rescaled proportionally to the new size of the image. After obtaining the predictions, the returned bounding boxes are transformed to match the original size of the image for the visualization and evaluation of the proposal results.

3.3 Training

TRG-Net uses two region proposal models, one for training and another for inference. The network is trained as the original Faster R-CNN, using an RPN to determine the regions of interest (Ren et al., 2015). However, during inference, TRG-Net uses a G-RPM, which provides regions of interest based on motion. Using two region proposal models helps to unify the detection and classification tasks without losing the strengths of each stage separately. The backbone, the last pooling and fully connected layers are not affected by the distinction between inference and training.

Using an RPN during training provides greater robustness and scope to the model, since using a G-RPM would cause loss of information. This is because the number of training objects is reduced when considering only moving objects. Furthermore, static objects in the training data could be dynamic in the test data, decreasing precision considerably.

The use of static images for training becomes more relevant when choosing a training data-set. To use a G-RPM, we would need the bounding boxes and labels of all the moving objects within a video. This data set would be too complex to obtain since motion detection is conventionally related to binary masks, and video object detection does not always look for moving objects. In fact, currently, there exists no public data set offering dense annotations for various complex scenes in video object detection (Zhu et al., 2020). In addition, using static images also broadens the applicability and flexibility of TRG-Net. Thanks to the fine-tuning of the well-known Faster R-CNN, our proposal could be applied to solve different real-world problems in various contexts.

4 EXPERIMENT ANALYSIS

Smart surveillance and autonomous driving are appropriate fields to apply the detection and classification of moving objects. Therefore, training and experiments were carried out using the kitti data set for 2D object detection (Geiger et al., 2012). This data-set consists of various images of urban environments geared toward autonomous driving. These images include seven main classes: ‘cars’, ‘vans’, ‘trucks’, ‘trams’, ‘pedestrians’, ‘people’, and ‘cyclists’. But also two additional classes: ‘misc’ and ‘dontCare’, useful to avoid overfitting and the collection of false positives.

The kitti data-set weighs 12 GB, and consists of 7481 training images and 7518 test images in PNG format, comprising a total of 80256 labeled objects. All the images are in RGB space. Although they do not have the same dimension, they have all been resized to $3 \times 1242 \times 375$. See Figure 3 for reference.

The labels include information about the object in the third dimension; however, for the handled 2D task the important features are:

- *type*: String determining the class of the object.
- *truncated*: Float between 0 (not truncated) and 1 (truncated), where truncated means that the boundaries of the object are not visible in the image.
- *occluded*: Integer indicating the status of the occlusion: 0 = fully visible, 1 = partially occluded, 2 = largely occluded, 3 = unknown.
- *alpha*: Float representing the angle of observation of the object, it goes from $-\pi$ to π.

Figure 3: Sample image of the kitti data-set with its labels.

Table 1: Comparison between TRG-Net and other object detection models. The more optimal values are highlighted with bold, and the less optimal are colored with red.

Model	Backbone	AP	# Parameters	Inference Time
TRG-Net (ours)	MobileNetV3	0.423	18.30 M	0.138 s
Faster R-CNN	MobileNetV3	0.423	18.91 M	0.221 s
	ResNet50	**0.519**	41.53 M	4.702 s
SSD Lite	MobileNetV3	0.283	**6.96 M**	**0.098 s**
RetinaNet	ResNet50	0.492	33.8 M	4.501 s

- *bbox*: Four floats $(x\_min, y\_min, x\_max, y\_max)$ determining the 2D bounding box of the object in the image.

Additionally, we calculated the object area using the *bbox* parameter. The tags *iscrowd* and *image_id* tags were also added to use the COCO metrics (Consortium, 2022), but they do not have real meaning for our evaluation.

4.1 Experimental Results

We employ two metrics to evaluate our proposal: (1) average precision (AP) and (2) inference time. AP refers to the average precision over multiple values of Intersection over Union (IoU) within a single image; this is the most representative measurement in object detection. And the inference time is the average forwarding step time over every image frame in a video. The detection and classification precision of moving objects cannot be strictly addressed due to the novel nature of the problem and the lack of a suitable benchmark, this issue will be discussed in detail in the following subsection 4.2.

The COCO evaluation method (Lin et al., 2014) is applied to calculate the AP. This interface computes the AP and recall values of a model over a test data set. On the other hand, the video frames were evaluated one by one due to the real-time nature of video surveillance cameras and autonomous driving systems. Thus, the inference time is calculated as the average response time of the model over consecutive video frames. That is, the input tensor for each test step contains a single image.

The experimental results obtained by TRG-Net and other object detection models are shown in Table 1. The models used to validate our proposal are: (1) Faster R-CNN (Ren et al., 2015) with MobileNetV3 (Howard et al., 2019), (2) Faster R-CNN with ResNet50 (He et al., 2016), (3) SSD Lite (Liu et al., 2016) with MobileNetV3, and (4) RetinaNet (Lin et al., 2017) with ResNet50. Our proposal is compared with object detection models because, to our knowledge, there are no similar proposals that detect and classify only moving objects. Thus, we are making the assumption that an object detector will identify moving objects in a subsequent task once it has identified all the objects within a frame.

The APs of the networks in table 1 were obtained after 10 training epochs of fine-tuning using the kitti data-set, we did not use data augmentation or any other data pre-processing technique to improve the AP values. The closer the AP value is to one, the higher the precision. Taking into account the use of MobileNetV3 as the backbone, the AP value of TRG-Net is higher than the AP value of SSD Lite. This is to be expected due to the two-stage nature of our proposal. However, the highest AP is achieved when using ResNet50, a deeper backbone. Note that the AP value is directly proportional to the number of parameters and inversely proportional to the inference time. Since precision and speed are required, a balance between both metrics should be sought.

Regarding the number of parameters of TRG-Net, this is low considering the difference in AP with respect to Faster R-CNN with ResNet50 and RetinaNet. See the detailed count of parameters in Table 2.

Table 2: Parameter count of TRG-Net modules.

Module	# Parameters
Backbone	4.36 M
RPN	0.61 M
Predictor	13.95
Total	18.91 M

While the lightweight backbone does not add much to the overall complexity, the last fully connected layers constitute most of the parameters and inference time. This is a common phenomenon in convolutional networks that perform object location and classification. During inference, TRG-Net does not consider the 0.61 million parameters of the RPN by replacing the convolutions with a GMM, thus reducing the number of parameters to 18.30 million.

We emulated the resources of a computationally

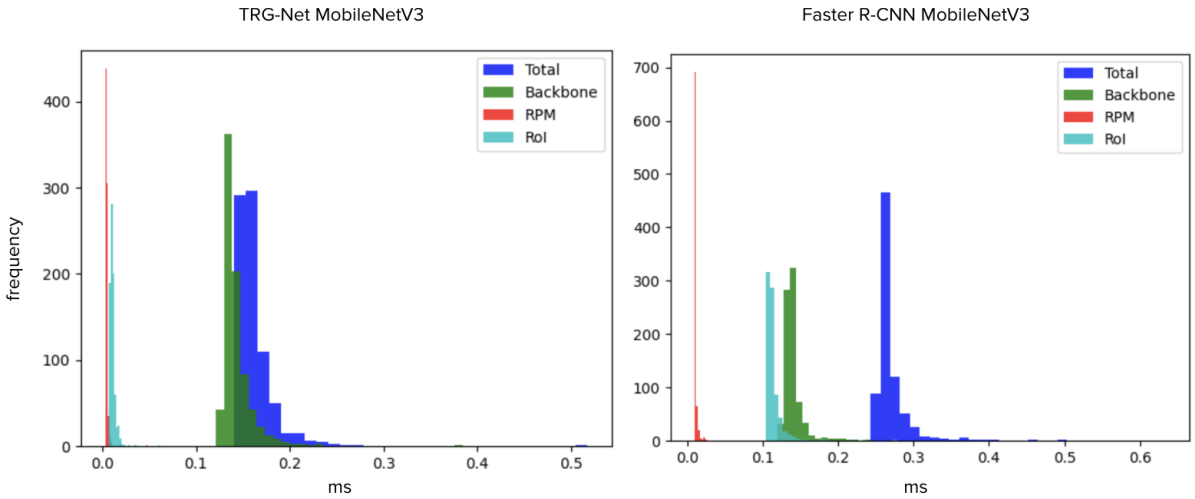

Figure 4: Histogram of the execution time by network module.

limited device to calculate the inference time. The device characteristics are the following: 4GB of memory and 0.5 CPUs of a 3.1 GHz Dual-Core Intel Core i5. The time values shown in table 1 were obtained by applying the models to a test video of 700 frames.

The time difference between Faster R-CNN with MobileNetV3 and TRG-net is due to the use of a traditional method to discover regions of interest. Avoiding the use of an RPN bypasses the calculations associated with its 0.6 million parameters. Nevertheless, the main timing gap is given by the reduced number of proposals when using G-RPM. This implies fewer operations since the network is computing smaller tensors. Note that SSD has the lowest inference time, this result is consistent with the number of parameters. However, such a speed sacrifices AP and the flexibility granted by a two-stage model. These times can be seen in Figure 4, a histogram dividing the execution time per module for TRG-Net and Faster R-CNN with MobileNetV3. The distribution of the values is uniform, and the standard deviation is low, which is good because it demonstrates stability during execution.

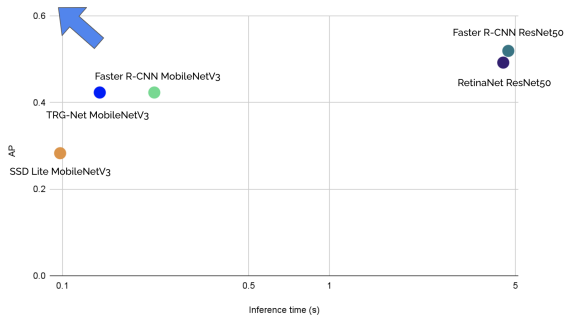

Figure 5: Distribution of the models according to their AP and inference time.

Figure 5 shows a global comparison considering the inference time and the AP simultaneously. The arrow points in the direction of the best setting that strikes a balance between speed and precision. Among this cloud of points, TRG-Net, Faster R-CNN with ResNet50, and the SSD Lite with MobileNetV3 stand out. SSD has the lowest inference time, in counterweight, its AP is also the lowest. Since SSD is a one-stage detector, it could not be adapted to the TRG-Net architecture, losing the benefits of G-RPM. On the other hand, using heavier backbones, such as ResNet50, increases the AP. However, it also increases the inference time, resulting in inadequate models for real-time solutions.

4.2 Discussion

After the execution of TRG-Net over several real street videos, we observed three things: (1) The precision and speed heavily depends on the G-RPM parameters, (2) videos with objects of heterogeneous sizes lower the performance of the network, and (3) detection is poor for videos captured by non-fixed cameras. By precision we do not refer to the evaluated AP, in this context precision means how well TRG-Net detects and classifies only moving objects. These observations show the proposal limitations, but also highlight the strengths of TRG-Net. Our model performs well when the correct G-RPM parameters are found, is fast if the objects have homogeneous sizes, and detection is good for videos captured by fixed cameras.

Figure 6 shows the output of the G-RPM with different values of learning rates lr, the minimum area parameter a was equal to 35. Note that the number of region proposals decreases when the lr value increases. That is, the higher lr, the fewer pixels will

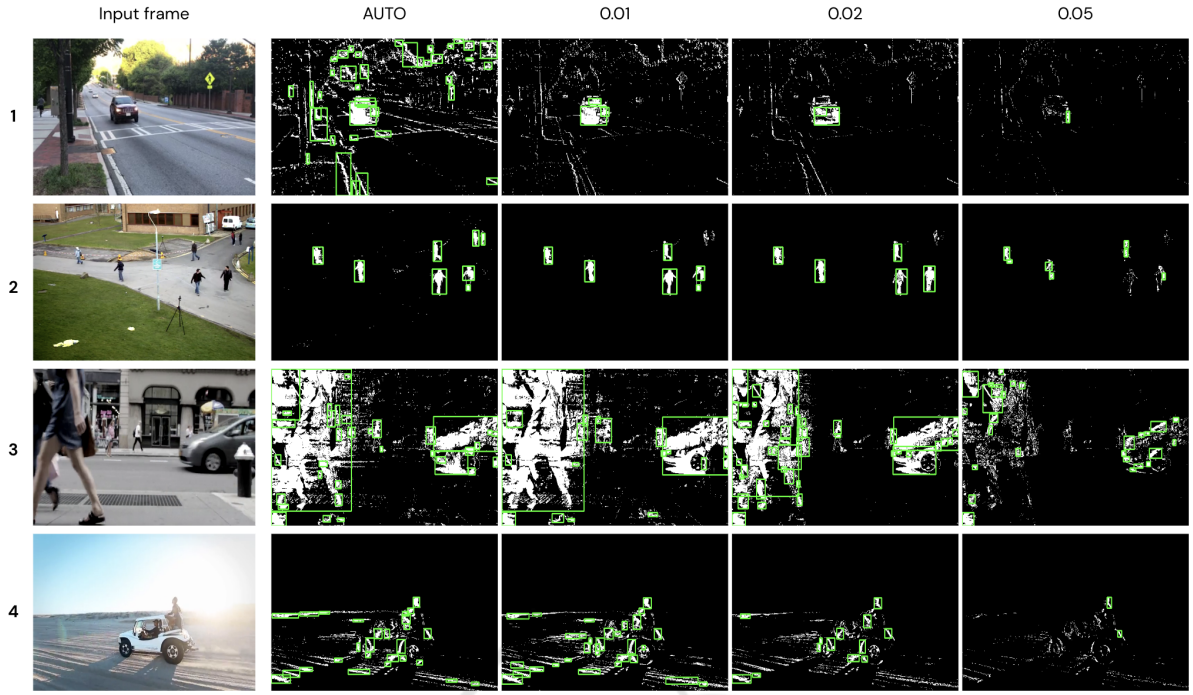

Figure 6: Visual comparison of G-RPM output with different learning rates over images with static and dynamic backgrounds.

be recognized as a change due to the speed of the GMM update. Video number 1 recognizes more objects when using a learning rate equal to 0.01, while video number 2 does it when using a learning rate equal to 0.02. Therefore, multiple values of lr should be tried to achieve the best results, but a value from 0.01 to 0.02 works fine in most cases.

Something similar occurs with the minimum-area parameter. Video number 2 has small objects, so an 35 value for a is correct. However, video number 3 has objects of heterogeneous sizes, so a very large value of a might miss small objects. A value close to 0 would include small objects, but also many irrelevant regions, especially if the video is noisy. Here, we can make a trade-off between precision and speed. Small values of a means a greater number of proposals, therefore, small objects will not be lost and the precision increases. On the other hand, as there are more proposals, the tensors will be larger and the speed of the network will be affected. It is not recommended to use values close to 0, it is better to find a value that represents the minimum size of the objects that you want to be detected.

Video number 4 was taken by a moving camera, thus, it has a dynamic background. One of the limitations of using a traditional MOD model, such as GMM, is its low performance when the background is not static. Since practically the entire scene is constantly changing, the regions proposed by G-RPM are very noisy, inaccurate, and incomplete. Therefore, TRG-Net can only be applied to videos captured by static cameras. Still, one point in favor of using GMM is that it deals with lighting changes; thus, our solution performs well in most videos with fixed backgrounds.

In addition to the three aforementioned points, we cannot bypass the novelty of the addressed problem. Literature has commonly dealt separately with video object detection, motion detection, and image classification. To our knowledge, detecting only moving objects has not been tackled yet, at least not explicitly and prioritizing the speed and precision of execution. This implies that there is no public data set benchmark to objectively measure the overall effectiveness of the solution, that is, the average AP of all the frames of a video considering only moving objects. The ImageNet VID data-set (Russakovsky et al., 2015) has perhaps the closest benchmark to compare our proposal, but this data-set contains annotations over static images and dynamic backgrounds. Therefore, it is not adequate to evaluate our proposal. In order to perform an accurate measurement of our model, only moving objects in videos should be labeled with bounding boxes and classes. Building such data-set is not a trivial task, and implies a complete study of the necessary resources and effort. Visually we can verify that TRG-Net detects and classifies moving objects, as can be seen in Figure 7. Therefore, we encourage

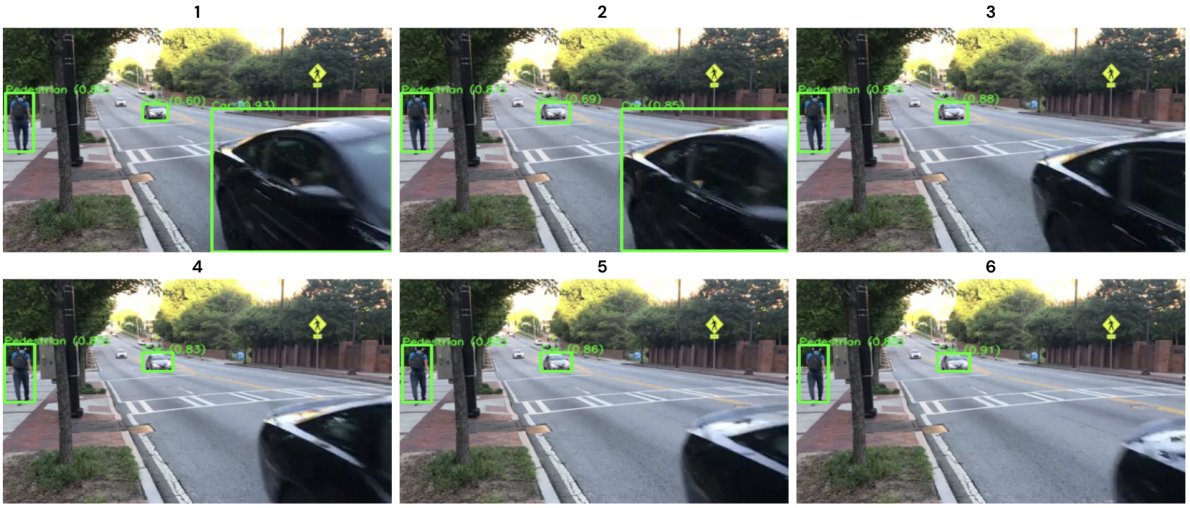

Figure 7: TRG-Net output showing the bounding box and class of moving objects in a video.

the creation of an adequate data-set and benchmark to be able to fully evaluate our proposal and future solutions.

Finally, we highlight the flexibility of our solution, since it could be extended to different contexts by modifying its components. Although TRG-Net is intended to be lightweight and run on computationally limited devices, we can increase classification precision by using deeper backbones, more complex GMMs, or by avoiding reducing the size of the input images. In the same way, we can prioritize speed by using much lighter backbones and adequate G-RPM parameters. Regarding its applicability, since the training is similar to the Faster R-CNN training, we can make use of pre-trained weights over different data sets, and just modify the use of G-RPM during inference.

5 CONCLUSIONS

We proposed TRG-Net, a lightweight GMM-based model that runs in near real time on computationally limited devices to address the fast detection and classification of moving objects. The evaluation showed that the AP is high compared to one-stage detectors using lightweight backbones. Regarding the model speed, the inference time is less compared to the Faster R-CNN using an RPN. Moreover, it can be reduced by applying lighter backbones and other values to the G-RPM parameters. A more complex GMM could be used to improve the G-RPM precision, as well as new heuristics to increase the number of proposals. Future work could evaluate the use of other methods for detecting moving objects, such as optical flow or even non-parametric models that avoid the task of choosing a learning rate and a minimum area for the G-RPM. Currently, there is no benchmark that measures the precision of classification and detection of moving objects at the same time. Thus, elaborating a proper validation data set and benchmark is a complex but necessary task. We are looking forward to TRG-Net being applied to solve real-world problems, such as smart surveillance systems, and set a precedent for unified detection and classification of moving objects.

REFERENCES

Albawi, S., Mohammed, T. A., and Al-Zawi, S. (2017). Understanding of a convolutional neural network. In *2017 international conference on engineering and technology (ICET)*, pages 1–6. Ieee.

Alsmirat, M. A., Obaidat, I., Jararweh, Y., and Al-Saleh, M. (2017). A security framework for cloud-based video surveillance system. *Multimedia Tools and Applications*, 76(21):22787–22802.

Braham, M., Pierard, S., and Van Droogenbroeck, M. (2017). Semantic background subtraction. In *2017 IEEE International Conference on Image Processing (ICIP)*, pages 4552–4556. Ieee.

Braham, M. and Van Droogenbroeck, M. (2016). Deep background subtraction with scene-specific convolutional neural networks. In *2016 International Conference on Systems, Signals and Image Processing (IWSSIP)*, pages 1–4. IEEE.

Chen, Y. and Hu, W. (2021). A video-based method with strong-robustness for vehicle detection and classification based on static appearance features and motion features. *IEEE Access*, 9:13083–13098.

Consortium, C. (2022). Detection evaluation.

Fan, L., Zhang, T., and Du, W. (2021). Optical-flow-based framework to boost video object detection performance with object enhancement. *Expert Systems with Applications*, 170:114544.

Geiger, A., Lenz, P., and Urtasun, R. (2012). Are we ready for autonomous driving? the kitti vision benchmark suite. In *2012 IEEE conference on computer vision and pattern recognition*, pages 3354–3361. IEEE.

Girshick, R., Donahue, J., Darrell, T., and Malik, J. (2014). Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 580–587.

Glegoła, W., Karpus, A., and Przybyłek, A. (2021). MobileNet family tailored for Raspberry Pi. *Procedia Computer Science*, 192:2249–2258.

Gruosso, M., Capece, N., and Erra, U. (2021). Human segmentation in surveillance video with deep learning. *Multimedia Tools and Applications*, 80(1):1175–1199.

Haouari, F., Faraj, R., and AlJa'am, J. M. (2018). Fog computing potentials, applications, and challenges. In *2018 International Conference on Computer and Applications (ICCA)*, pages 399–406. IEEE.

He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778.

Hou, B., Liu, Y., Ling, N., Liu, L., and Ren, Y. (2021). A Fast Lightweight 3D Separable Convolutional Neural Network With Multi-Input Multi-Output for Moving Object Detection. *IEEE Access*, 9:148433–148448.

Howard, A., Sandler, M., Chu, G., Chen, L.-C., Chen, B., Tan, M., Wang, W., Zhu, Y., Pang, R., Vasudevan, V., et al. (2019). Searching for mobilenetv3. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1314–1324.

Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., and Adam, H. (2017). MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications.

Hung, M.-H., Pan, J.-S., and Hsieh, C.-H. (2014). A fast algorithm of temporal median filter for background subtraction. *J. Inf. Hiding Multim. Signal Process.*, 5(1):33–40.

Jagannathan, P., Rajkumar, S., Frnda, J., Divakarachari, P. B., and Subramani, P. (2021). Moving Vehicle Detection and Classification Using Gaussian Mixture Model and Ensemble Deep Learning Technique. *Wireless Communications and Mobile Computing*, 2021:1–15.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25.

Kulchandani, J. S. and Dangarwala, K. J. (2015). Moving object detection: Review of recent research trends. In *2015 International Conference on Pervasive Computing (ICPC)*, pages 1–5. IEEE.

Lin, T.-Y., Goyal, P., Girshick, R., He, K., and Dollár, P. (2017). Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2980–2988.

Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., and Zitnick, C. L. (2014). Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer.

Liu, L., Ouyang, W., Wang, X., Fieguth, P., Chen, J., Liu, X., and Pietikäinen, M. (2020). Deep Learning for Generic Object Detection: A Survey. *International Journal of Computer Vision*, 128(2):261–318.

Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.-Y., and Berg, A. C. (2016). Ssd: Single shot multibox detector. In *European conference on computer vision*, pages 21–37. Springer.

Machado, P., Oikonomou, A., Ferreira, J. F., and Mcginnity, T. M. (2021). HSMD: An Object Motion Detection Algorithm Using a Hybrid Spiking Neural Network Architecture. *IEEE Access*, 9:125258–125268.

Patel, H. A. and Thakore, D. G. (2013). Moving object tracking using kalman filter. *International Journal of Computer Science and Mobile Computing*, 2(4):326–332.

Redmon, J., Divvala, S., Girshick, R., and Farhadi, A. (2016). You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788.

Ren, S., He, K., Girshick, R., and Sun, J. (2015). Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, 28.

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., et al. (2015). Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3):211–252.

Simonyan, K. and Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.

Song, Z., Ali, S., and Bouguila, N. (2020). Background subtraction using infinite asymmetric gaussian mixture models with simultaneous feature selection. *IET Image Processing*, 14(11):2321–2332.

Suzuki, S. et al. (1985). Topological structural analysis of digitized binary images by border following. *Computer vision, graphics, and image processing*, 30(1):32–46.

Tan, M. and Le, Q. (2019). Efficientnet: Rethinking model scaling for convolutional neural networks. In *International conference on machine learning*, pages 6105–6114. PMLR.

Tayeb, A. A., Aldhaheri, R. W., and Hanif, M. S. (2021). Vehicle speed estimation using gaussian mixture model and kalman filter. *International Journal of Computers, Communications and Control*, 16(4).

Xie, S., Girshick, R., Dollár, P., Tu, Z., and He, K. (2017). Aggregated residual transformations for deep neural

networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1492–1500.

Yang, F., Karanam, S., Zheng, M., Chen, T., Ling, H., and Wu, Z. (2022). Multi-motion and Appearance Self-Supervised Moving Object Detection. In *2022 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 2101–2110. IEEE.

Yang, Y., Loquercio, A., Scaramuzza, D., and Soatto, S. (2019). Unsupervised Moving Object Detection via Contextual Information Separation. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 879–888. IEEE.

Yi, Z. and Liangzhong, F. (2010). Moving object detection based on running average background and temporal difference. In *2010 IEEE International Conference on Intelligent Systems and Knowledge Engineering*, pages 270–272. IEEE.

Yuan, Y. (2021). A pet detection system based on yolov4. In *2021 2nd International Seminar on Artificial Intelligence, Networking and Information Technology (AINIT)*, pages 342–348. IEEE.

Zaidi, S. S. A., Ansari, M. S., Aslam, A., Kanwal, N., Asghar, M., and Lee, B. (2022). A survey of modern deep learning based object detection models. *Digital Signal Processing*, page 103514.

Zhang, J. (2021). Navigating in the clouds: The triumphs and drawbacks of the cloud act.

Zhao, Z.-Q., Zheng, P., Xu, S.-T., and Wu, X. (2019). Object Detection With Deep Learning: A Review. *IEEE Transactions on Neural Networks and Learning Systems*, 30(11):3212–3232.

Zhu, H., Wei, H., Li, B., Yuan, X., and Kehtarnavaz, N. (2020). A review of video object detection: Datasets, metrics and methods. *Applied Sciences*, 10(21):7834.

Zivkovic, Z. (2004). Improved adaptive gaussian mixture model for background subtraction. In *Proceedings of the 17th International Conference on Pattern Recognition, 2004. ICPR 2004.*, volume 2, pages 28–31. IEEE.