# Human pose estimation for mitigating false negatives in weapon detection in video-surveillance

Alberto Lamas[a,*], Siham Tabik[a], Antonio Cano Montes[c], Francisco Pérez-Hernández[a], Jorge García, Roberto Olmos[a,b], Francisco Herrera[a]

[a] *Dpt. of Computer Science and Artificial Intelligence, Andalusian Research Institute in Data Science and Computational Intelligence, DaSCI, University of Granada, 18071, Granada, Spain*
[b] *National Supercomputer Laboratory of Southeast Mexico, Meritorious Autonomous University of Puebla (BUAP), Puebla, Mexico*
[c] *Biomedical research foundation, San Carlos Clinical Hospital, Madrid, Spain*

## Abstract

Applying CNN-based object detection models to the task of weapon detection in video-surveillance is still producing a high number of false negatives. In this context, most existing works focus on one type of weapons, mainly firearms, and improve the detection using different pre- and post-processing strategies. One interesting approach that has not been explored in depth yet is the exploitation of the human pose information for improving weapon detection. This paper proposes a top-down methodology that first determines the hand regions guided by the human pose estimation then analyzes those regions using a weapon detection model. For an optimal localization of each hand region, we defined a new factor, called *Adaptive pose factor*, that takes into account the distance of the body from the camera. Our experiments show that this top-down Weapon Detection over Pose Estimation (WeDePE) methodology is more robust than the alternative bottom-up approach and state-of-the art detection models in both indoor and outdoor video-surveillance scenarios.

*Keywords:* Weapon detection, human pose estimation, object detection in videos, video-surveillance, real-time object detection.

[*] Corresponding author

*Email addresses:* albertocl@ugr.es (Alberto Lamas ), siham@ugr.es (Siham Tabik), antonio.cano.montes@hotmail.com (Antonio Cano Montes), fperezhernandez@ugr.es (Francisco Pérez-Hernández), herrera@decsai.ugr.es (Francisco Herrera)

## 1. Introduction

It is unquestionable that the simple presence of a weapon, for example a handgun or a knife, in different video-surveillance scenarios generates a situation of danger. If the weapon is held by a person, the situation becomes more dangerous and requires an urgent security response. Reformulating the problem of weapons detection into the detection of a weapon held by a person can surely reduce the space of the information to be analyzed and hence minimize the detection errors.

Most existing weapon detection solutions focus only on the detection of firearms in indoor video-surveillance. In particular, they select one of the most relevant object detection models, train it on a custom dataset then, apply different pre-processing [1][2] and post-processing [3][4] techniques to further improve the detection. The proposed approaches mainly search for isolated firearms in each frame without considering the presence of humans in the scene. Very few studies tried to exploit the human presence; however the proposed solutions are not reproducible and do not provide neither models nor dataset, which make their evaluation and comparison impossible [5][6][7].

To exploit the presence of one or multiple persons in the scene, this work propose reducing the search area from the entire frame into the regions that contain the hands of those persons. Then analyze only those areas of interest using a weapon detection model. This approach has a high potential for reducing the number of False Negatives (FN) and False Positives (FP).

This work presents the top-down Weapon Detection over Pose Estimation (WeDePE) methodology which is reproducible and traceable, that exploits the human presence in scenarios in which a person is carrying a weapon, firearm or knife. Our objective is mitigating FN of hardly visible weapons as well as reducing FP in the background. The human presence information is expressed using the pose estimation to later determine the regions where a hand is located in the frame. These hand regions will be the regions of interest to detect a weapon.

The designed Top-down WeDePE methodology follows the next stages:

(a) The human pose estimation model analyses the input frame and estimates the pose of each person in the scene.

(b) The hand regions of each person are localized based on the computed coordinates and the optimal size of each hand region is calculated using a new factor named *Adaptive pose factor*.

(c) All the hand regions are extracted and used to build a new single image.

(d) The weapon detection model analyses the new generated image and outputs the set of hand-regions that are considered as weapon with high-confidence.

It is also analyzed and compared the performance of this approach with two approaches: (i) a bottom-up approach that separately finds a hand and a weapon in the input frame to later compare whether they are located in the same region of the image, and (ii) a single weapon detector that analyses the entire frame looking for a weapon.

This paper is organized as follows. Section 2 depicts the pose estimation process and object detection models that are implemented in the Top-down WeDePE methodology and reviews the most related works on weapon detection and along with pose estimation. Section 3 describe the Top-down WeDePE methodology for weapon detection guided by human pose estimation. Section 4 provide the experimental analyses that shows the potential of the proposed top-down methodology. Finally, Section 5 summarizes the conclusions and future work.

## 2. Preliminaries

This section provides a brief description of the background required to understand the proposed Top-down WeDePE methodology: human pose estimation models (Section 2.1) and single image object detection models (Section 2.2) then related work (Section 2.3).

### *2.1. Human pose estimation*

Human pose estimation refers to the task of localizing human joints also called key-points, e.g., right elbow, left elbow, right wrist, left wrist and so on, in images or videos. A human pose estimation model analyzes an input image that contains one or multiple persons and output the coordinates $(x, y)$ of a maximum of eighteen body joints. The detected number of key-points depends strongly on the quality of the input image, whether part of the body is hidden and the distance at which the person is situated with respect to the camera.

DeepPose model [8] was the first approach in using Convolutional Neural Networks (CNN) [9] based regression models to estimate body joints. This approach provided good estimations by that time but with poor generalization capacity. A subsequent approach [10] reformulated the problem to estimating a set of heatmaps; each map indicates the location confidence of a certain key-point. The output is a discrete heatmap instead of continuous regression. Heatmaps provided better estimates than the regression-based approach, however, they lack structure modeling.

An important advance was achieved by Convolutional Pose Machines [11], an end-to-end model that organizes the joint estimation into layers. This model outperformed previous methods obtaining 87.95% PCKh-0.5 (Probability of the Correct Keypoint with Head lenght reference) on MPII database [12]. The last architectural optimizations have produced important improvements, up to 92.3% [13] [14]. New feature representation (Part Affinity Fields) [15] allows faster bottom-up approaches (from a cloud of joints to pose estimation of different persons), that culminated in the real time multi-person pose estimation system OpenPose [16].

### *2.2. Single image object detection models*

In general, single-image object detection models can be classified into two groups single-stage and two-stage detectors.

**Two-stage detectors:** This category of detection models is represented by Region based CNN (R-CNN) [17] and its subsequent optimizations, namely

Fast R-CNN [18] and Faster R-CNN [19]. This type of detectors operates in two different stages, the first stage selects the possible areas or regions of the image that may contain the objects of interest and the second stage analyses these candidate regions with a CNN-classification model to determine whether they contain the searched object of interest.

This type of detectors is more robust especially on higher resolution images and provide better detections than the single-stage detectors; although they are more expensive computationally.

**One-stage detectors:** This type of detectors directly seeks to predict the position of the object and the class to which it belongs in a single-stage. The most important examples are EfficientDet [20], RetinaNet [21], CenterNet [22], SSD [23], and YOLO [24] [25]. Single-stage algorithms can reach very high frame rate processing on GPUs and due to their low computational requirements, some of them can run on edge computing devices at speeds close to 20 FPS. Usually these algorithms use lower image resolutions and can have difficulties detecting small objects.

### *2.3. Related work to weapon detection in video surveillance*

Related works to weapon detection in video-surveillance can be broadly divided into two groups. Those that improve the detection by building new training datasets and utilizing the state-of-the art single-image object detection models to videos [26] [27] [28] and those that improve the detection by applying different pre-processing [1] [2] and post-processing [3] [4] optimizations including data and model fusion [5] [7][6].

Most works focus on one specific type of weapon, either pistol or knife. The seminal work in this context is [26], in which the authors built the first firearm dataset and an alarm system that analyzes the input videos using Faster R-CNN and triggers an alarm when a pistol is detected in five consecutive frames. The quality of the detection decreases in low quality videos, i.e., low contrast and presence of blur. Subsequently, several works proposed improving the detection by increasing the size of the training dataset using synthetic images [28] or images acquired from a Closed Circuit TeleVision (CCTV) setup [27].

Several works proposed pre-processing techniques that helps reducing the number of FP and FN in the detection. The authors in [1] proposed a brightness and contrast correction pre-processing technique to improve detrimental light reflection produced by metallic weapons. The authors in [2] proposed a binocular vision based pre-processing technique to eliminate the background and hence reduce FP and FN. Alternatively, the authors in [3] improved the weapon detection by analyzing the output of the detection, i.e., predicted regions, using an auto-encoder model. While the authors in [4] showed that applying a binary classification method on the detected regions improves the overall performance.

Very few works considered the presence of persons for improving weapon detection in videos [5] [6] [7]. The authors in [5] first, apply a person detection model then analyze the obtained predicted bounding boxes (bbox) using a firearm detection model. The most related works to ours are [6] [7].

In [6], the authors used the human skeletal pose estimate to detect the threat in each frame. They designed a multi-stage classification model, a first CNN determines whether a person and a handgun are present in an image. If so, a second CNN estimates the pose of the person and finally a feed-forward neural network assesses the threat level based on the joint positions of the persons skeletal pose estimate from the previous stage. The main drawback of this approach is that it does not perform a detection task, it only classifies individual frames.

In [7], the authors first estimate the hand regions based the pose information then, jointly analyze the two-halves normalized binary pose image and the hand regions using a classification model. The authors stated that this approach provides better overall performance than the one-stage detector YOLOv3 alone. Unfortunately, this approach is not reproducible as the given description does not include all the important details. Last but not the least, the fact that neither the dataset nor the models are available makes the comparison with our approach impossible.

The present work is different to all the previous works as it provides a complete and reproducible approach together with a deep experimental analysis in both indoor and outdoor scenarios. Besides, we aim at detecting firearms as well as knives, while previous works focus on one single type of weapons, mainly firearms.

## 3. Top-down Weapon Detection over Pose Estimation methodology

The top-down approach first extracts the hand regions from the input frame based on the pose information then analyses these regions using a weapon detection model. The proposed Top-down WeDePE methodology requires a good human pose estimation and hence a precise estimation of the hand regions, i.e., the areas determined by the elbow and wrist joints.

This section present the approach for estimating the region of hands based on the Adaptive hand regions estimation method in Section 3.1, and the hand regions image generation procedure to reduce the computational cost of the weapon detection stage in Section 3.2. Finally, the flowchart of the Top-down WeDePE methodology is depicted in Section 3.3.

### *3.1. Adaptive hand regions estimation method*

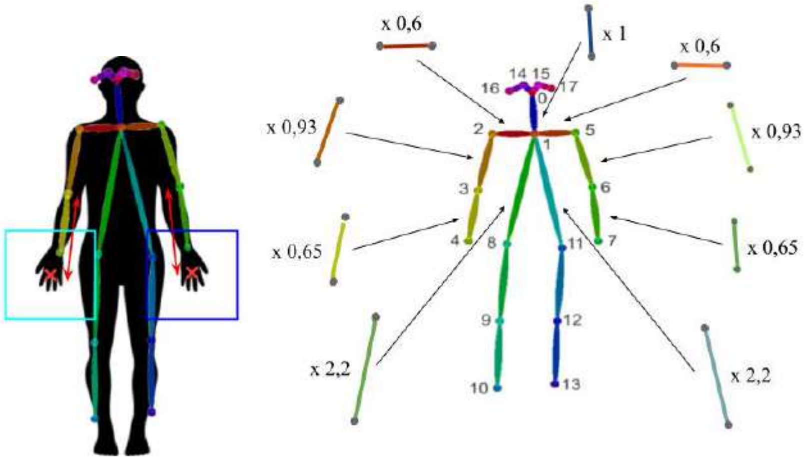
The localization of the hand and the estimation of the square area that surrounds it, is critical as it must include all the necessary information for its further analysis using a weapon detection model. That is, if part of the weapon is eliminated, especially in the case of a knife, this will be more likely to produce a FN since the tip of the knife is the most important feature for its correct detection.

In addition, the quality of the pose estimation and hence the quality of the hand localization depends on the distance between the person and the camera. The farthest the person from the camera, the more challenging is the estimation as parts of the body can be either occluded or blurry. To take all these aspects into account, the new factor, named *Adaptive Pose factor*, determines the optimal size of the square region that surrounds each hand.

In particular, we use the state-of-the art pose estimation model to calculate all the pose key-points. Then, we calculate the coordinates of the hand $(x_H, y_H)$ in the image (Equation 1) based on the position of the elbow and wrist as follows:

$$\begin{aligned} x_H &= x_W + 0.5 * (x_W - x_E) \\ y_H &= y_W + 0.5 * (y_W - y_E) \end{aligned} \tag{1}$$

Where $_H$, $_E$, $_W$ refers respectively to hand, elbow and wrist. The value 0.5 corresponds to the ratio of elbow-wrist limb to get the hand coordinates.



a) Hand coordinates estimation b) Limbs and leg-to-body ratios used for *Adaptive Pose factor*

Figure 1: Illustration of (a) the hand coordinates estimation and (b) the leg-to-body ratios used to compute the *Adaptive Pose factor* [29] [30].

Finally, the *length* of the square centered in the hand (See Figure 1 (a)) is based on the proposed Adaptive pose factor (Equation 2), that modifies the size of the region according to the position of the person in the scene and a subset of the more stable limbs shown in Figure 1. It is calculated as follows:

$$Adaptive\ Pose\ factor = \sum_{i=0}^{N} \frac{L_i \times \mathrm{ratio}_i}{N} \tag{2}$$

$$\mathrm{length} = \ Adaptive\ Pose\ factor * 1.2$$

The *Adaptive Pose factor* is calculated using the leg-to-body ratio information provided in [29] and validated in [30] (see Figure 1 (b)). Where $N$ is the total number of segments determined by the pose model, $L$ is the length of each limb and *ratio* is the ratio leg-to-body. The value 1.2 is a parameter determined experimentally in a way that most weapons fit in the square centred in the hand. An illustration can be seen in Figure 2.

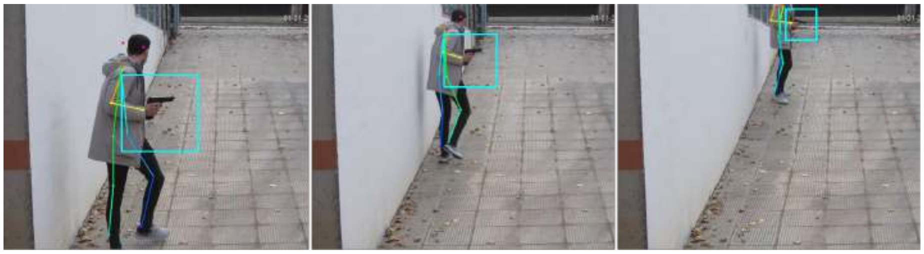


Figure 2: Illustration of how the *Adaptive Pose Factor* adapts the hand regions (i.e., the bounding box in blue color) at different distances from the camera. The further the smaller.

### *3.2. Hand regions image generation procedure*

The presence of several bodies in a scene implies the analysis of a large number of hand regions. To reduce the computational cost of this processing, the proposed Top-down WeDePE methodology includes an additional optimization before the weapon detection stage. For each input frame a new single image of the same size is generated using all the hand regions detected in the input-frame (see Figure 3). The new generated image can be seen as a grid of cell-images. The number of cells increases with the detected hands in the input frame.

Given the estimated hand regions, the number of regions determines the structure of the grid of $R \times C$ regions of the same size, where the number

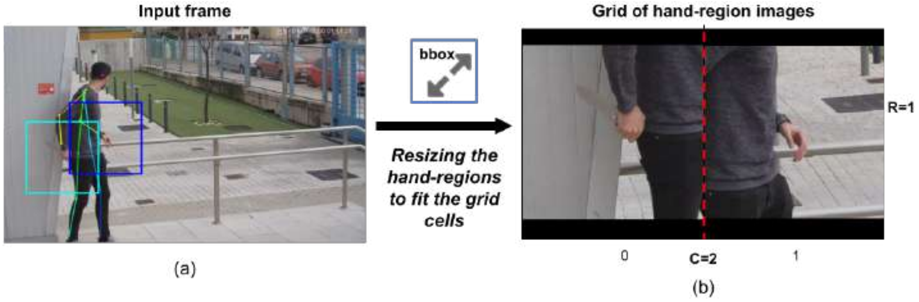

Input frame
bbox
Resizing the hand-regions to fit the grid cells
Grid of hand-region images
R=1
0
C=2
1
(a)
(b)

Figure 3: The two hand-regions detected in input frame (a) are used to compose a new single image of the same size (b). The new image is actually a 1 × 2 hand-regions grid.

of rows $R$ and columns $C$ are defined following the pseudo code below (see Algorithm 1).

**Algorithm 1** Generate a new image, which is actually a 2 × 1grid of images, filled by the extracted hand regions

```
procedure HAND-REGIONS-IMAGE(hands_number)
    number of cols $C \leftarrow 2$
    number of rows $R \leftarrow 1$
    while $C \times R < hands\_number$ do
        if $C < (R \times 2)$ then
            $C \leftarrow C + 1$
        else
            $R \leftarrow R + 1$
```

The size of the grid is calculated based on Algorithm 1 and is filled with all the estimated hand regions extracted from the input frame. Each hand region is placed in a cell of the grid. For instance, in the example illustrated in Figure 3, the two estimated hand regions fill a 1 × 2 grid. Those hand regions are resized to fill a grid of the same size as the input-frame. This makes the handled objects larger, more visible, and with less distracting information or objects.

### *3.3. Flowchart of the Top-down WeDePE methodology*

The field of view monitored by a video surveillance camera includes a vast amount of information. Traditionally, all the information is processed by the detector of the weapon detection system seeking weapons such as a knife or a pistol. Nevertheless, the weapon detection can be reformulated as a much simpler process. First, detecting the presence of persons in the scene then analyzing only the hand regions for searching possible handled weapons. This way, it reduces the amount of background information and increase the size of analyzed regions.

The Top-down WeDePE methodology addresses this hypothesis by using the pose information and estimated hand regions as unique regions of interest in the image to detect weapons.

Top-down WeDePE methodology is designed to be reproducible and traceable. Its different integrated modules combine the information as shown in Figure 4, and follows the next steps:

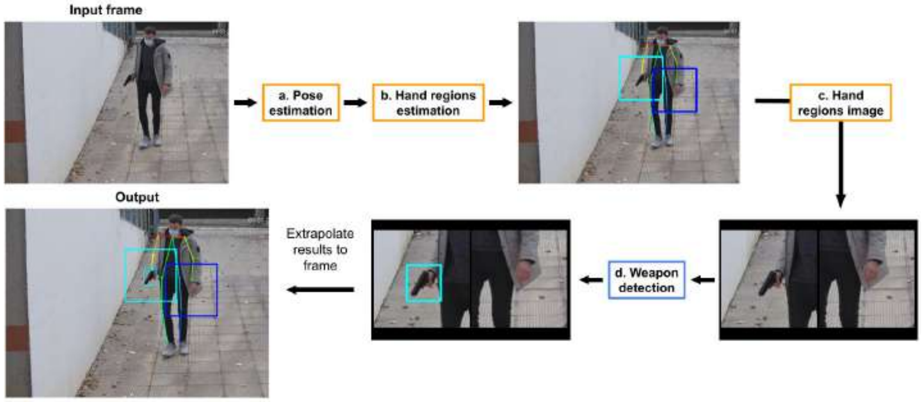


Figure 4: Illustration of the flowchart of the Top-down WeDePE methodology for weapon detection.

(a) Pose estimation: the input frame is analyzed using a pose estimation model to compute the pose information of all the underlying bodies. This information is actually a set of 2d-coordinates associated to each body.

(b) Hand regions estimation: the Adaptive hand regions estimation method is applied for each human pose information. The hand localization is calculated using the direction vector of the elbow-wrist limbs, and the size of the square region that delimits each hand region is estimated using the Adaptive pose factor.

(c) Hand regions image: the square hand regions are extracted from the underlying frame and a new image is created following the pre-processing described in Hand regions image construction procedure. The generated image has the same size as the input frame and it is built as a grid of images. Each cell of the grid is occupied by a hand-region image resized to fill the cell space.

(d) Weapon detection: the weapon detection model analyzes the new hand regions image and outputs the detected weapons with a high confident threshold value.

## 4. Experimental analysis

In this section we evaluate and compare the proposed top-down methodology with a bottom-down WeDePE methodology and also with several state-of-the art detection models trained for weapon detection.

The description of the common experimental points is provided in Section 4.1. The analysis and comparison of the top-down and bottom-up approaches, and single detectors is provided in Section 4.2. An illustrative analysis of the FN and FP is provided in Section 4.3 and 4.4 respectively.

### *4.1. Experimental setup*

This section details the common considerations for the experiments as follows. The detection models and implementation details are described in Section 4.1.1. Metrics to compare performance are elaborated in Section 4.1.2. The dataset and test videos are depicted in Section 4.1.3. Lastly, the flowchart illustration of the Bottom-up WeDePE methodology in Section 4.1.4.

#### *4.1.1. Deep Learning models for object detection*

The four selected detection models for evaluating the proposed approaches include the two types of detection architectures. The two-stage detector Faster R-CNN [19] based on ResNet101, and different one-stage detectors such as SSD [23] based on ResNet50, EfficientDet [10] based on D3, and CenterNet [22] based on Hourglass104. All the models were pretrained on COCO dataset and fine-tuned on our dataset for weapons detection using the default hyperparameters, which are available in [1].

The human pose estimation model that provides the estimated coordinates of the body points for the experiments is the pre-trained OpenPose model developed and provided in [15]. For a the estimation of the key-points of the body, we used a confidence threshold value of 0.5. The size of the input image is 328 × 328 pixels.

All the implementation were performed using TensorFlow 2 [31].

### *4.1.2. Evaluation metrics*

To evaluate and compare the performance of the detection and Top-down WeDePE methodology as weapon detection systems at the frame level, we used standard mean average precision (mAP) (Eq. 3) averaging IoU range from 0.5 to 0.95 in 0.05 steps and single 0.5 IoU level.

$$mAP = \frac{\sum_{i=1}^{K} AP_i}{K} \qquad AP_i = \frac{1}{10} \sum_{r \in [0.5, \ldots, 0.95]} \int_0^1 p(r) dr \tag{3}$$

where given $K$ classes (knife, pistol), $p$ precision and r *recall* define $p(r)$ as the area under the interpolated precision-recall curve for class $i$.

We also used metric Precision, Recall, and F1 score (Eq. 4) to evaluate detected regions (confidence over 50%) and positive detection (IoU over 50%) and comparable to mAP [0.5].

$$precision = \frac{TP}{TP + FP}, \quad recall = \frac{TP}{TP + FN}, \quad F1 = 2 \times \frac{precision \times recall}{precision + recall} \tag{4}$$

[1] github.com/tensorflow/models/blob/master/research/object_detection/g3doc/tf2_detection_zoo.md

Where True Positive (TP) refers to the number of weapons correctly detected in the frames of the input video. One or more bbox that correctly detect a weapon are considered once. FP refers to the number of bbox produced by the detection model in which there is no weapon or IoU less than 50% with the ground truth. FN refers to the total number of visible weapons not correctly detected.

### *4.1.3. Dataset*

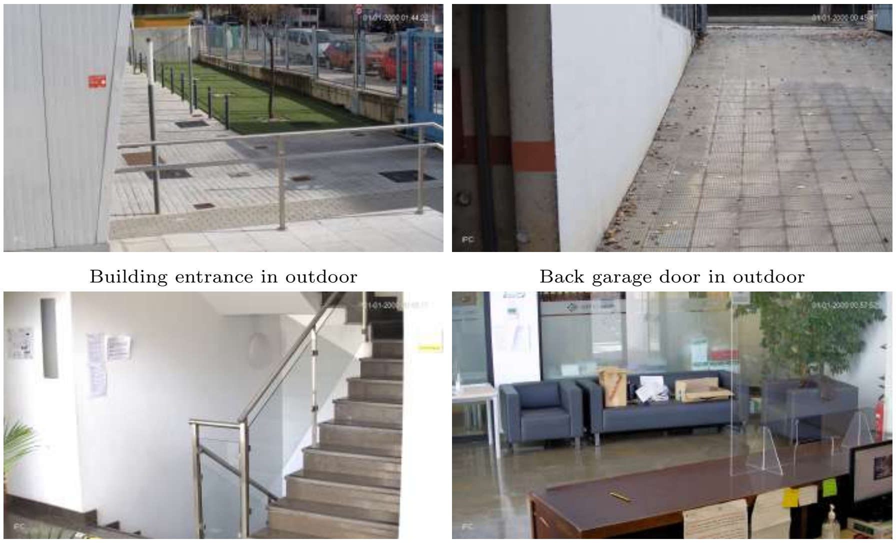
The weapon detection models have been trained on the weapon classes, knives and pistols, provided in *Sohas weapon detection* dataset [4][2]. The total number of images are 3250, where the knife class includes 1825 images and the pistol class includes 1425 images.

***Test set.*** The evaluation of the bottom-up and Top-down WeDePE methodology against the single detection model has been performed using fifteen videos recorded in four different scenarios. The building entrance and back garage door in outdoor environment, a transit area in indoor and service desk in indoor. These scenarios are depicted in Figure 5.

The fifteen test videos show weapons such as knifes or pistols in the four scenarios. Each scenario presents different challenges as describe below:

- Building entrance from outdoor zone (video 1-4): Outdoor scenario where a person moves from farther away to closer a position in a complex background.
- Back garage door from outdoor zone (video 5-8): Outdoor scenario where a person moves up to a large distance from the camera on a slightly difficult background.
- Service desk indoor (video 9-10): Indoor scenario where a person moves through a passageway, and the frontal angle and distance make complex the weapon detection on the stairs.

[2] https://dasci.es/transferencia/open-data/24705/



Building entrance in outdoor

Back garage door in outdoor

Transit area in indoor

Service desk in indoor

Figure 5: Example frame of the four scenarios used in the fifteen test videos.

- Transit area indoor (video 11-15): Indoor scenario where a person stands behind the desk, and the background include many objects and reflections.
- Indoor transit area with several people located at different distances from the camera (video 16).

### *4.1.4. Bottom-up weapon detection approach combined with pose estimation*

For comparison purposes, we have developed a bottom-up weapon detection approach combining pose estimation and weapon detection models that addresses potential detection errors in the background. It shares the a, b, and c stages of the top-down fusion methodology as illustrated in Figure 6.

The bottom-up approach follows the next steps:

(a) The weapon detection model analyzes each input frame and outputs a number of candidate detections with a high confident threshold value.

(b) If there is a positive weapon detection, the underlying frame is analyzed using the pose estimation model, which computes the pose information (joint and limbs).

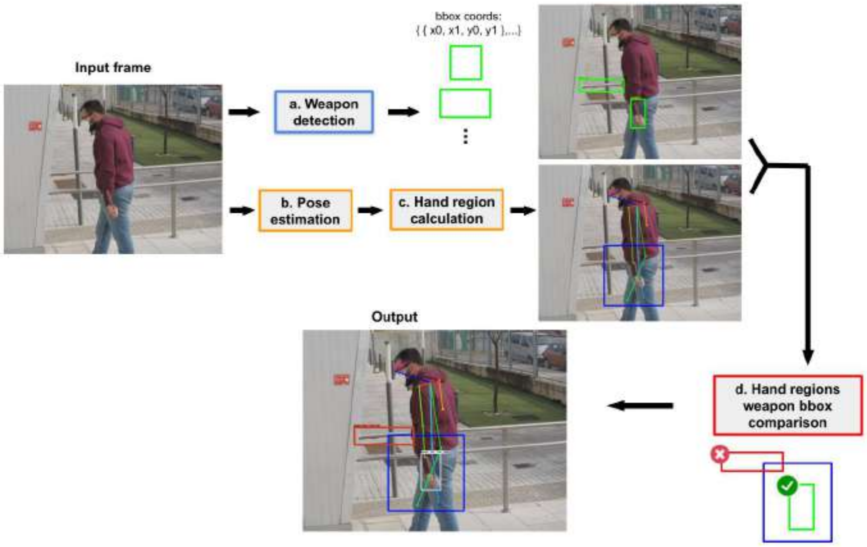

bbox coords:
{ { x0, x1, y0, y1 },...}
Input frame
a. Weapon detection
b. Pose estimation
c. Hand region calculation
Output
d. Hand regions weapon bbox comparison

Figure 6: Illustration of the flowchart of the Bottom-up WeDePE methodology for weapon detection.. Discarded regions are painted in red in the output frame.

(c) The hand localization is calculated using formula (1). Then length of the square region centred in the hand is estimated using formula (2).

(d) The region of each positive detection is compared with all hand regions based only on their coordinates. If the Intersection over Union (IoU) is higher than a threshold value, the detection is validated. The regions out of hand regions are discarded.

### *4.2. Performance analysis of the Top-down WeDePE methodology in the task of weapon detection in video-surveillance*

We carried out a comparison between the proposed top-down and the Bottom-up WeDePE approaches (described in Section 4.1) and several single detection models over the four considered scenarios, two indoor and two outdoor. The performance in terms of precision, recall and F1 averaged over the four considered scenarios is shown in Table 1.

The top-down and Bottom-up WeDePE approaches outperform in general the single detection model-based solution in all the scenarios and videos (Analysis per video is provided in Appendix).

| **Zone** | **Scene** | **Approach** | **Precision**(%) | **Recall**(%) | **F1**(%) |
|---|---|---|---|---|---|
| | Building entry door outdoor | Single detector | 79.5 | 57.0 | 66.4 |
| Out | 1529 frames | Top-down | **91.2** | **87.1** | **89.1** |
| | | Bottom-up | 80.8 | 56.9 | 66.7 |
| | Back garage door outdoor | Single detector | 83.1 | 63.1 | 71.8 |
| Out | 1140 frames | Top-down | **96.2** | **83.8** | **89.6** |
| | | Bottom-up | 85.0 | 61.0 | 71.1 |
| | Service desk indoor | Single detector | 73.2 | 59.1 | 65.4 |
| In | 626 frames | Top-down | **88.8** | **62.9** | **73.6** |
| | | Bottom-up | 76.8 | 58.0 | 66.1 |
| | Transit area indoor | Single detector | 60.5 | 55.5 | 57.9 |
| In | 1729 frames | Top-down | **90.1** | **84.1** | **87.0** |
| | | Bottom-up | 61.8 | 55.5 | 58.5 |
| | | Single detector | 74.1 | 58.7 | 65.4 |
| | Averaged | Top-down | **91.6** | **79.5** | **84.8** |
| | | Bottom-up | 76.1 | 57.9 | 65.6 |

Table 1: Comparison between the bottom-up, top-down and single detection model in four scenarios, two indoor (In) and two outdoor (Out).

It particular, the Top-down WeDePE methodology provides remarkable performance with respect to the bottom-up approach with up to 29.6%, 30.1%, and 29.1% improvement respectively in precision, recall and F1, in all scenarios. The lowest performance of the top-down approach was obtained in the *Service desk indoor* scenario. This is due to the fact that this scenario includes more detrimental conditions for weapon detection and pose estimation such as several important parts of the body are hidden by a desk and Covid-19 protection screen.

Besides, we analyzed the impact of single-stage and two-stage detection models on the performance of top-down and bottom-up using the fifteen test videos. Table 2 shows the precision, recall and F1 when including Faster R-CNN, SSD, EfficientDet and CenterNet as weapon detection stage into the top-down and bottom-up approaches. The performance of single detection models is also included for comparison purposes.

The top-down and bottom-up approaches provide better precision, recall, and F1 independently on the used detection model. In particular, top-down approach overcomes the bottom-up one. This improvement is more impressive when including one-stage detectors instead of two-stage detector, i.e., Faster

| **Included detector** | **Approach** | **Precision**(%) | **Recall**(%) | **F1**(%) | **Frame rate (fps)** |
|---|---|---|---|---|---|
| FasterR-CNN | Single FasterR-CNN | 87.0 | 74.0 | 80.0 | 10.36 ± 0.43 |
| | Bottom-up | **88.4** | 73.4 | 80.2 | 9.51 ± 0.37 |
| | Top-down | 85.7 | **83.8** | **84.7** | 8.89 ± 0.61 |
| SSD | Single SSD | 71.0 | 61.2 | 65.8 | 18.63 ± 0.67 |
| | Bottom-up | 71.4 | 60.6 | 65.6 | 15.38 ± 0.68 |
| | Top-down | **92.9** | **79.1** | **85.4** | 16.07 ± 0.57 |
| EfficientDet | Single EfficientDet | 67.7 | 69.1 | 68.4 | 17.38 ± 0.56 |
| | Bottom-up | 67.8 | 67.6 | 67.7 | 15.48 ± 0.53 |
| | Top-down | **94.4** | **91.5** | **92.9** | 15.08 ± 0.45 |
| CenterNet | Single CenterNet | 67.5 | 29.4 | 40.9 | 15.86 ± 0.6 |
| | Bottom-up | 72.6 | 29.1 | 41.6 | 13.52 ± 0.51 |
| | Top-down | **94.5** | **73.6** | **82.8** | 13.27 ± 0.4 |

Table 2: The performance of the top-down and bottom-up with different detection models over all scenarios.

R-CNN; with a boost of up to 27% in precision, 44.2% in recall, and 41.9% in F1. This means that the top-down approach becomes a much better weapon detector when including a one-stage detector, especially EfficientDet. A more detailed analysis of all the approaches on each one of the fifteen test videos is provided in the Appendix section [3]. In addition, from Table 1 and 2, it can be observed that the top-down approach, on average, reduces the frame rate by 14.8% with respect to single detector while maintaining in all cases a general performance improvement of 19.4%.

*Remark: Why mAP is not appropriate for evaluating the Top-down WeDePE approach for weapon detection?*

As one can observe from Table 3, the bottom-up approach provides higher mAP and mAP with IoU in 0.5 but with much lower precision, recall and F1 than the top-down approach. These higher mAP values are actually due to the fact that Stage (d) in the bottom-up approach filters a large number of inaccurate candidate regions and validates only few candidate regions, which improves the value of mAP. However, Stage (d) in the top-down approach generates a large number of candidate regions that are very close from the weapon or ground truth. This is due to the fact that the input image to that stage contains much

[3] Videos available in: youtube.com/playlist

less background and hence increases the number of potential incorrect candidate regions that are not finally considered as TP. The Appendix shows more details in terms of TP, FN and FP in all the test videos.

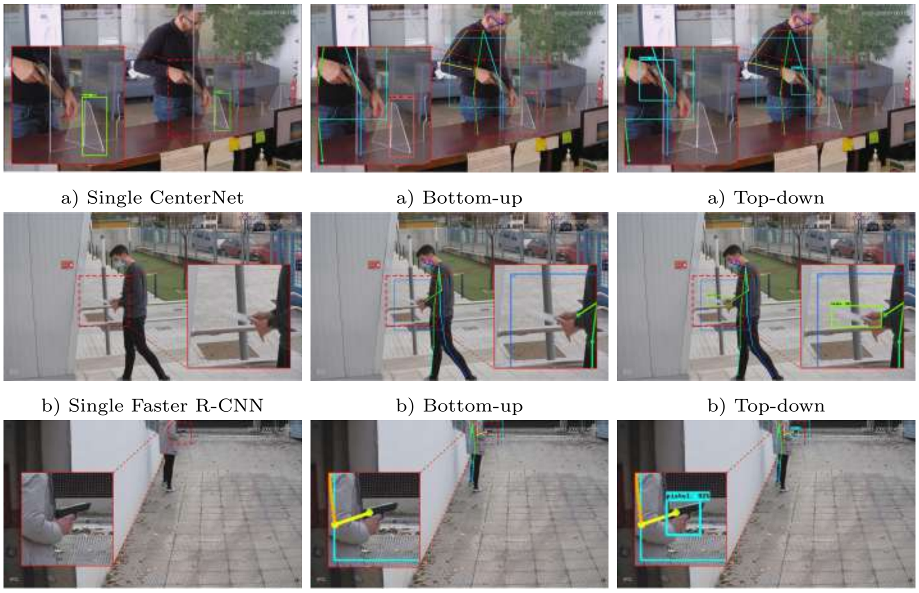
| | **Scene** | **Approach** | **mAP** | **mAP[0.5]** | **Precision**(%) | **Recall**(%) | **F1**(%) |
|---|---|---|---|---|---|---|---|
| video 1-15 | Averaged | Single detector | 38.22 | 72.00 | 74.1 | 58.7 | 65.4 |
| | | Top-down | 37.49 | 64.95 | **91.6** | **79.5** | **84.8** |
| | | Bottom-up | **40.72** | **77.26** | 76.1 | 57.9 | 65.6 |

Table 3: Comparison between the bottom-up, top-down and the corresponding single detection model over the fifteen test videos. This results illustrate the inappropriateness of mAP metric for evaluating the task of weapon detection.

Therefore, this distortion produced in the mAP makes more reliable to evaluate the quality of the weapon detection task using the precision, recall and F1 metrics.

### *4.3. Illustrative analysis of false negative mitigation*

After an exhaustive analysis of the results obtained by the Top-down WeDePE methodology on the fifteen test videos we found out that there exist three types of FN or weapons that were not detected as illustrated in Figure 7.



a) Single CenterNet a) Bottom-up a) Top-down

b) Single Faster R-CNN b) Bottom-up b) Top-down

c) Single CenterNet c) Bottom-up c) Top-down

Figure 7: Examples of FN recovered by the Top-down WeDePE methodology. The frames are extracted from the scenarios *Service desk indoor* frame a), *Building entry door outdoor* frame b), and *Back garage door outdoor* frame c). The region outlined by the red lines shows the area of interest when zoomed in. Color code of bbox for single detection and top-down approaches shows green-knife and blue-pistols, for bottom-up approach red-discarded detection and white-valid weapon.

The first type of common FN occurs with clearly visible weapons behind the Covid-19 protective screen producing a considerable number of miss-detections in this part of the image, an example is shown in Figure 7a), where the FN pistol is not detected by the single detection model and bottom-up approach but correctly detected by the top-down approach.

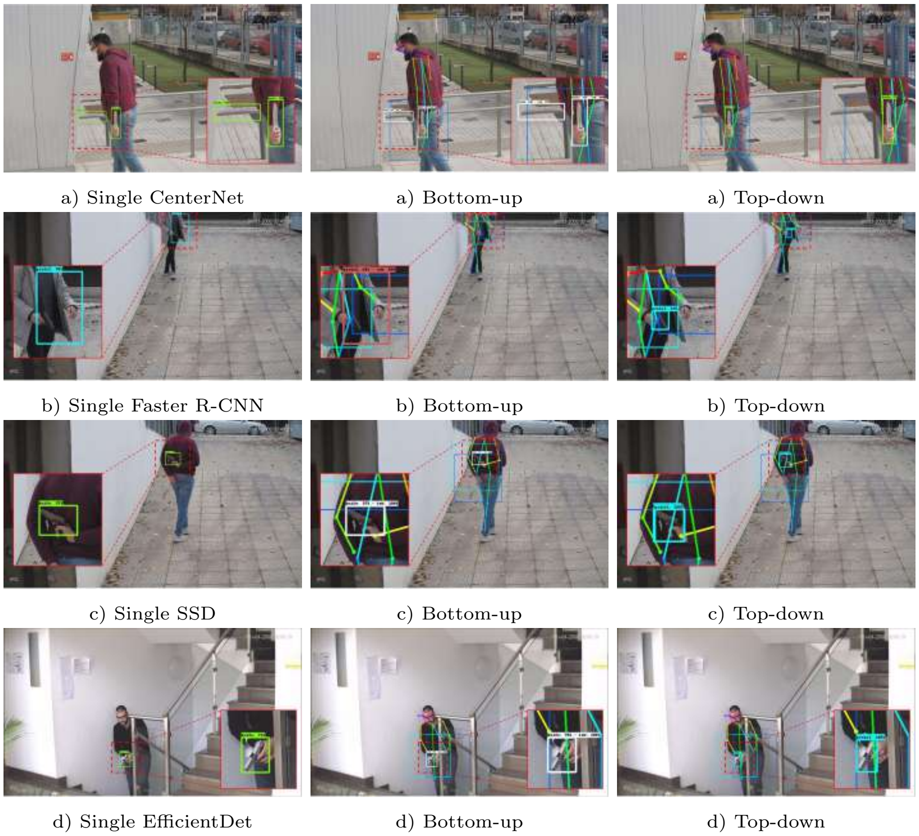
The second type of FN is also very common and occurs when the weapon and background do not produce enough contrast as shown in Figure 7b), where the weapon shape is blurred and it can only be detected thanks to the clearer view in the hand region image from the top-down approach.

The third type of FN occur when the weapon and the person the held that weapon are far away from the camera, see an example in Figure 7c), where the pistol is too small in the image to be located by region proposal algorithms (more pronounced difficulty for one-stage detectors) however the top-down approach improves the detection of weapons at larger distances.

We have also found that weapons under certain conditions in outdoor scenarios can be barely visible. Diversity of features in the background and increased exposure to environmental conditions make difficult both the weapon detection and image capturing by camera. Which reduces the amount of background information and increases the size of the handled object and hence improves the detection capability of top-down approach as shown in Figure 7.

### *4.4. Illustrative analysis of false positive corrections*

The process of FN mitigation carried out by the Top-down WeDePE methodology improves in parallel the detection of some type of FP, especially when a weapon is confused with other objects in the background. An illustration of this type of FP that occurs in outdoor scenarios or under challenging conditions is illustrated in Figure 8.



a) Single CenterNet a) Bottom-up a) Top-down

b) Single Faster R-CNN b) Bottom-up b) Top-down

c) Single SSD c) Bottom-up c) Top-down

d) Single EfficientDet d) Bottom-up d) Top-down

Figure 8: Examples of type of FP corrected by bottom-up and Top-down WeDePE approach. The frames are extracted from the scenarios *Building entry door outdoor* frame a), *Back garage door outdoor* frame b), *Back garage door outdoor* frame c), and *Transit area indoor* frame d). The region outlined by the red lines shows the area of interest when zoomed in. Color code of bbox for single detection and top-down approaches shows green-knife and blue-pistols, for bottom-up approach red-discarded detection and white-valid weapon.

The first type occurs with metallic objects near the hand area or when the weapon is located at a considerable distance to the camera, see examples in Figure 8a) and d), the FP are validated in the bottom-up approach but are corrected in the top-down approach thanks to a clearer view of the objects in the hand region image.

The second type of FP occurs when a weapon on a challenging background is not properly adjusted by the region proposal stage showing big and off-centered bbox, see an example in Figure 8b), the regions remains the same in the bottom-up approach, but the pistol is correctly detected by the top-down approach.

Finally, other type of standard detection errors occurs with profile view of weapons where the characteristic shape of the weapon is lost, see an example in Figure 8c), the pistol against the clothing and slightly in profile is detected but miss-classified then the bottom-up approach can not correct the FP unlike the top-down approach detecting it despite the detrimental conditions.

## 5. Conclusions and future works

This work presented a Top-down WeDePE methodology that exploits the human pose estimation for mitigating FN in the detection of weapons, firearms and knives, held by a person in video-surveillance. The proposed methodology uses the key-points produced by the human pose estimation model to localize the hand regions in the frame. To estimate the optimal size of the hand-regions that will be analyzed by the weapon detection stage, we defined a new factor named Adaptive pose factor. The experiments showed that the top-down approach improves the detection performance, with respect to a bottom-up approach, by up to 17.5% precision, 20.8% recall, and 19.4% F1 score in the fifteen analysed videos in different scenarios.

As the proposed top-down approach depends on the human pose estimation, it can be combined with a single weapon detector to build a more robust CCTV system since the latter can detect weapons that are not necessarily held by a human.

As future work, we are planning to integrate the combination of different weapon detection methodologies for different purposes [26][1][2][4] [32] together with the Top-down WeDePE methodology on a CCTV system with the objective to guarantee the detection of weapons in all the situations and scenarios.

## Acknowledgments

This work was supported by regional projects, A-TIC-458-UGR18 (DeepL-ISCO) and P18-FR-4961 (BigDDL-CET), and national projects AI_MARS CIEN and PID2020-119478GB-I00. S. Tabik was supported by the Ramon y Cajal Programme (RYC-2015-18136).

## References

[1] A. Castillo, S. Tabik, F. Pérez, R. Olmos, F. Herrera, Brightness guided preprocessing for automatic cold steel weapon detection in surveillance videos with deep learning, Neurocomputing 330 (2019) 151–161.

[2] R. Olmos, S. Tabik, A. Lamas, F. Perez-Hernandez, F. Herrera, A binocular image fusion approach for minimizing false positives in handgun detection with deep learning, Information Fusion 49 (2019) 271–280.

[3] N. Vallez, A. Velasco-Mata, O. Deniz, Deep autoencoder for false positive reduction in handgun detection, Neural Computing and Applications 33 (2020) 1–11.

[4] F. Pérez-Hernández, S. Tabik, A. Lamas, R. Olmos, H. Fujita, F. Herrera, Object detection binary classifiers methodology based on deep learning to identify small objects handled similarly: Application in video surveillance, Knowledge-Based Systems 194 (2020) 105590.

[5] D. Romero, C. Salamea, Convolutional models for the detection of firearms in surveillance videos, Applied Sciences 9 (15) (2019) 2965.

[6] B. Abruzzo, K. Carey, C. Lowrance, E. Sturzinger, R. Arnold, C. Korpela, Cascaded neural networks for identification and posture-based threat assessment of armed people, in: 2019 IEEE International Symposium on Technologies for Homeland Security (HST), 2019, pp. 1–7.

[7] J. Ruiz-Santaquiteria, A. Velasco-Mata, N. Vallez, G. Bueno, J. A. Alvarez, O. Deniz, Handgun detection using combined human pose and weapon appearance, arXiv preprint arXiv:2010.13753.

[8] A. Toshev, C. Szegedy, Deeppose: Human pose estimation via deep neural networks, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2014, pp. 1653–1660.

[9] Y. LeCun, Y. Bengio, G. Hinton, Deep learning, nature 521 (7553) (2015) 436–444.

[10] J. Tompson, R. Goroshin, A. Jain, Y. LeCun, C. Bregler, Efficient object localization using convolutional networks, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2015, pp. 648–656.

[11] S.-E. Wei, V. Ramakrishna, T. Kanade, Y. Sheikh, Convolutional pose machines, in: Proceedings of the IEEE conference on Computer Vision and Pattern Recognition, 2016, pp. 4724–4732.

[12] M. Andriluka, L. Pishchulin, P. Gehler, B. Schiele, 2d human pose estimation: New benchmark and state of the art analysis, in: Proceedings of the IEEE Conference on computer Vision and Pattern Recognition, 2014, pp. 3686–3693.

[13] A. Newell, K. Yang, J. Deng, Stacked hourglass networks for human pose estimation, in: European conference on computer vision, Springer, 2016, pp. 483–499.

[14] K. Sun, B. Xiao, D. Liu, J. Wang, Deep high-resolution representation learning for human pose estimation, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2019, pp. 5693–5703.

[15] Z. Cao, T. Simon, S.-E. Wei, Y. Sheikh, Realtime multi-person 2d pose estimation using part affinity fields, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2017, pp. 7291–7299.

[16] Z. Cao, G. Hidalgo, T. Simon, S.-E. Wei, Y. Sheikh, Openpose: realtime multi-person 2d pose estimation using part affinity fields, IEEE transactions on pattern analysis and machine intelligence 43 (2019) 172–186.

[17] R. Girshick, J. Donahue, T. Darrell, J. Malik, Region-based convolutional networks for accurate object detection and segmentation, IEEE transactions on pattern analysis and machine intelligence 38 (2015) 142–158.

[18] R. Girshick, Fast r-cnn, in: Proceedings of the IEEE international conference on computer vision, 2015, pp. 1440–1448.

[19] S. Ren, K. He, R. Girshick, J. Sun, Faster r-cnn: Towards real-time object detection with region proposal networks, arXiv preprint arXiv:1506.01497.

[20] M. Tan, R. Pang, Q. V. Le, Efficientdet: Scalable and efficient object detection, in: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2020, pp. 10781–10790.

[21] T.-Y. Lin, P. Goyal, R. Girshick, K. He, P. Dollár, Focal loss for dense object detection, in: Proceedings of the IEEE international conference on computer vision, 2017, pp. 2980–2988.

[22] K. Duan, S. Bai, L. Xie, H. Qi, Q. Huang, Q. Tian, Centernet: Keypoint triplets for object detection, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2019, pp. 6569–6578.

[23] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, A. C. Berg, Ssd: Single shot multibox detector, in: European conference on computer vision, Springer, 2016, pp. 21–37.

[24] J. Redmon, S. Divvala, R. Girshick, A. Farhadi, You only look once: Unified, real-time object detection, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 779–788.

[25] A. Bochkovskiy, C.-Y. Wang, H.-Y. M. Liao, Yolov4: Optimal speed and accuracy of object detection, arXiv preprint arXiv:2004.10934.

[26] R. Olmos, S. Tabik, F. Herrera, Automatic handgun detection alarm in videos using deep learning, Neurocomputing 275 (2018) 66–72.

[27] J. L. S. González, C. Zaccaro, J. A. Álvarez-García, L. M. S. Morillo, F. S. Caparrini, Real-time gun detection in cctv: An open problem, Neural networks 132 (2020) 297–308.

[28] N. Vallez, A. Velasco-Mata, J. J. Corroto, O. Deniz, Weapon detection for particular scenarios using deep learning, in: Iberian Conference on Pattern Recognition and Image Analysis, Springer, 2019, pp. 371–382.

[29] T. M. Versluys, R. A. Foley, W. J. Skylark, The influence of leg-to-body ratio, arm-to-body ratio and intra-limb ratio on male human attractiveness, Royal Society open science 5 (2018) 171790.

[30] B. Bogin, M. I. Varela-Silva, Leg length, body proportion, and health: a review with a note on beauty, International journal of environmental research and public health 7 (2010) 1047–1075.

[31] M. Abadi, A. Agarwal, P. Barham, et al., TensorFlow: Large-scale machine learning on heterogeneous systems, software available from tensorflow.org (2015).
URL https://www.tensorflow.org/

[32] R. Olmos, S. Tabik, F. Perez-Hernandez, A. Lamas, F. Herrera, Multicast: Multi confirmation-level alarm system based on cnn and lstm to mitigate false alarms for handgun detection in video-surveillance, arXiv preprint arXiv:2104.11653.

## 6. Appendix

This section provides the performance results of the bottom-up, top-down, and the corresponding single detector approaches using four different detection models on fifteen test videos (Table 4-19) in the four video surveillance scenarios as describe in the Section 4.1.3. #TP, #FP and #FN in Tables 4 to 19 refers respectively to the number of TP, FP and FN. The averaged results on each video are provided in Table 20.

| **Detector** | **Approach** | **#TP** | **#FP** | **#FN** | **Precision**(%) | **Recall**(%) | **F1**(%) |
|---|---|---|---|---|---|---|---|
| FasterR-CNN | Single Faster R-CNN | 370 | 29 | 45 | 92.7 | 89.2 | 90.9 |
| | Top-down | 391 | 106 | 24 | 78.7 | **94.2** | 85.7 |
| | Bottom-up | 367 | 14 | 48 | **96.3** | 88.4 | **92.2** |
| SSD | Single SSD | 293 | 24 | 122 | 92.4 | 70.6 | 80.1 |
| | Top-down | 300 | 7 | 115 | **97.7** | **72.3** | **83.1** |
| | Bottom-up | 290 | 24 | 125 | 92.4 | 69.9 | 79.6 |
| EfficientDet | Single EfficientDet | 295 | 7 | 120 | **97.7** | 71.1 | 82.3 |
| | Top-down | 377 | 17 | 38 | 95.7 | **90.8** | **93.2** |
| | Bottom-up | 293 | 7 | 122 | **97.7** | 70.6 | 82.0 |
| CenterNet | Single CenterNet | 204 | 22 | 211 | 90.3 | 49.2 | 63.7 |
| | Top-down | 274 | 12 | 141 | 95.8 | **66.0** | **78.2** |
| | Bottom-up | 204 | 7 | 211 | **96.7** | 49.2 | 65.2 |

Table 4: Performance comparison between the bottom-up, top-down and corresponding single detection model in video 1.

| **Detector** | **Approach** | **#TP** | **#FP** | **#FN** | **Precision**(%) | **Recall**(%) | **F1**(%) |
|---|---|---|---|---|---|---|---|
| FasterR-CNN | Single Faster R-CNN | 406 | 10 | 139 | **97.6** | 74.5 | **84.5** |
| | Top-down | 473 | 161 | 72 | 74.6 | **86.8** | 80.2 |
| | Bottom-up | 406 | 10 | 139 | **97.6** | 74.5 | **84.5** |
| SSD | Single SSD | 430 | 2 | 115 | **99.5** | 78.9 | **88.0** |
| | Top-down | 449 | 29 | 96 | 93.9 | **82.4** | 87.8 |
| | Bottom-up | 430 | 2 | 115 | **99.5** | 78.9 | **88.0** |
| EfficientDet | Single EfficientDet | 427 | 24 | 118 | 94.7 | 78.3 | 85.7 |
| | Top-down | 480 | 19 | 65 | **96.2** | **88.1** | **92.0** |
| | Bottom-up | 427 | 24 | 118 | 94.7 | 78.3 | 85.7 |
| CenterNet | Single CenterNet | 286 | 77 | 259 | 78.8 | 52.5 | 63.0 |
| | Top-down | 394 | 5 | 151 | **98.7** | **72.3** | **83.5** |
| | Bottom-up | 286 | 38 | 259 | 88.3 | 52.5 | 65.8 |

Table 5: Performance comparison between the bottom-up, top-down and corresponding single detection model in Video 2.

| **Detector** | **Approach** | **#TP** | **#FP** | **#FN** | **Precision**(%) | **Recall**(%) | **F1**(%) |
|---|---|---|---|---|---|---|---|
| FasterR-CNN | Single Faster R-CNN | 216 | 24 | 77 | **90.0** | 73.7 | 81.1 |
| | Top-down | 267 | 72 | 26 | 78.8 | **91.1** | **84.5** |
| | Bottom-up | 216 | 24 | 77 | **90.0** | 73.7 | 81.1 |
| SSD | Single SSD | 60 | 55 | 233 | 52.2 | 20.5 | 29.4 |
| | Top-down | 286 | 2 | 7 | **99.3** | **97.6** | **98.5** |
| | Bottom-up | 60 | 55 | 233 | 52.2 | 20.5 | 29.4 |
| EfficientDet | Single EfficientDet | 94 | 166 | 199 | 36.2 | 32.1 | 34.0 |
| | Top-down | 286 | 5 | 7 | **98.3** | **97.6** | **97.9** |
| | Bottom-up | 94 | 163 | 199 | 36.6 | 32.1 | 34.2 |
| CenterNet | Single CenterNet | 15 | 0 | 278 | 100.0 | 5.1 | 9.7 |
| | Top-down | 269 | 0 | 24 | 100.0 | **91.8** | **95.7** |
| | Bottom-up | 15 | 0 | 278 | 100.0 | 5.1 | 9.7 |

Table 6: Performance comparison between the bottom-up, top-down and corresponding single detection model in Video 3.

| **Detector** | **Approach** | **#TP** | **#FP** | **#FN** | **Precision**(%) | **Recall**(%) | **F1**(%) |
|---|---|---|---|---|---|---|---|
| FasterR-CNN | Single Faster R-CNN | 238 | 17 | 38 | **93.3** | 86.2 | **89.6** |
| | Top-down | 250 | 55 | 26 | 82.0 | **90.6** | 86.1 |
| | Bottom-up | 238 | 17 | 38 | **93.3** | 86.2 | **89.6** |
| SSD | Single SSD | 142 | 108 | 134 | 56.8 | 51.4 | 54.0 |
| | Top-down | 233 | 41 | 43 | **85.0** | **84.4** | **84.7** |
| | Bottom-up | 142 | 108 | 134 | 56.8 | 51.4 | 54.0 |
| EfficientDet | Single EfficientDet | 166 | 110 | 110 | 60.1 | 60.1 | 60.1 |
| | Top-down | 262 | 29 | 14 | **90.0** | **94.9** | **92.4** |
| | Bottom-up | 166 | 110 | 110 | 60.1 | 60.1 | 60.1 |
| CenterNet | Single CenterNet | 50 | 74 | 226 | 40.3 | 18.1 | 25.0 |
| | Top-down | 257 | 14 | 19 | **94.8** | **93.1** | **94.0** |
| | Bottom-up | 50 | 74 | 226 | 40.3 | 18.1 | 25.0 |

Table 7: Performance comparison between the bottom-up, top-down and single detection model in Video 4.

| **Detector** | **Approach** | **#TP** | **#FP** | **#FN** | **Precision**(%) | **Recall**(%) | **F1**(%) |
|---|---|---|---|---|---|---|---|
| FasterR-CNN | Single Faster R-CNN | 226 | 17 | 93 | 93.0 | 70.8 | 80.4 |
| | Top-down | 240 | 24 | 79 | 90.9 | **75.2** | **82.3** |
| | Bottom-up | 211 | 5 | 108 | **97.7** | 66.1 | 78.9 |
| SSD | Single SSD | 137 | 41 | 182 | 77.0 | 42.9 | 55.1 |
| | Top-down | 235 | 0 | 84 | **100** | **73.7** | **84.8** |
| | Bottom-up | 137 | 41 | 182 | 77.0 | 42.9 | 55.1 |
| EfficientDet | Single EfficientDet | 161 | 125 | 158 | 56.3 | 50.5 | 53.2 |
| | Top-down | 237 | 50 | 82 | **82.6** | **74.3** | **78.2** |
| | Bottom-up | 146 | 122 | 173 | 54.5 | 45.8 | 49.7 |
| CenterNet | Single CenterNet | 91 | 7 | 228 | 92.9 | 28.5 | 43.6 |
| | Top-down | 283 | 14 | 36 | 95.3 | **88.7** | **91.9** |
| | Bottom-up | 91 | 0 | 228 | **100** | 28.5 | 44.4 |

Table 8: Performance comparison between the bottom-up, top-down and corresponding single detection model in Video 5.

| **Detector** | **Approach** | **#TP** | **#FP** | **#FN** | **Precision**(%) | **Recall**(%) | **F1**(%) |
|---|---|---|---|---|---|---|---|
| FasterR-CNN | Single Faster R-CNN | 211 | 41 | 29 | 83.7 | 87.9 | 85.8 |
| | Top-down | 240 | 0 | 0 | **100** | **100** | **100** |
| | Bottom-up | 211 | 41 | 29 | 83.7 | 87.9 | 85.8 |
| SSD | Single SSD | 178 | 178 | 62 | 50.0 | 74.2 | 59.7 |
| | Top-down | 233 | 12 | 7 | **95.1** | **97.1** | **96.1** |
| | Bottom-up | 178 | 178 | 62 | 50.0 | 74.2 | 59.7 |
| EfficientDet | Single EfficientDet | 58 | 235 | 182 | 19.8 | 24.2 | 21.8 |
| | Top-down | 240 | 5 | 0 | **98.0** | **100** | **99.0** |
| | Bottom-up | 58 | 235 | 182 | 19.8 | 24.2 | 21.8 |
| CenterNet | Single CenterNet | 120 | 36 | 120 | 76.9 | 50.0 | 60.6 |
| | Top-down | 226 | 12 | 14 | **95.0** | **94.2** | **94.6** |
| | Bottom-up | 120 | 12 | 120 | 90.9 | 50.0 | 64.5 |

Table 9: Performance comparison between the bottom-up, top-down and corresponding single detection model in Video 6.

| **Detector** | **Approach** | **#TP** | **#FP** | **#FN** | **Precision**(%) | **Recall**(%) | **F1**(%) |
|---|---|---|---|---|---|---|---|
| FasterR-CNN | Single Faster R-CNN | 259 | 7 | 65 | **97.4** | 79.9 | 87.8 |
| | Top-down | 293 | 14 | 31 | 95.4 | **90.4** | **92.9** |
| | Bottom-up | 259 | 7 | 65 | **97.4** | 79.9 | 87.8 |
| SSD | Single SSD | 257 | 5 | 67 | 98.1 | 29.3 | 87.7 |
| | Top-down | 254 | 0 | 70 | **100** | **78.4** | **87.9** |
| | Bottom-up | 254 | 2 | 70 | 99.2 | **78.4** | 87.6 |
| EfficientDet | Single EfficientDet | 288 | 10 | 36 | 96.6 | 88.9 | 92.6 |
| | Top-down | 302 | 17 | 22 | 94.7 | **93.2** | **93.9** |
| | Bottom-up | 286 | 7 | 38 | **99.2** | 78.4 | 87.6 |
| CenterNet | Single CenterNet | 180 | 2 | 144 | **98.9** | 55.6 | 71.1 |
| | Top-down | 233 | 7 | 91 | 97.1 | **71.9** | **82.6** |
| | Bottom-up | 180 | 2 | 144 | **98.9** | 55.6 | 71.1 |

Table 10: Performance comparison between the bottom-up, top-down and corresponding single detection model in Video 7.

| **Detector** | **Approach** | **#TP** | **#FP** | **#FN** | **Precision**(%) | **Recall**(%) | **F1**(%) |
|---|---|---|---|---|---|---|---|
| FasterR-CNN | Single Faster R-CNN | 199 | 0 | 58 | **100** | 77.4 | 87.3 |
| | Top-down | 209 | 2 | 48 | 99.1 | **81.3** | **89.3** |
| | Bottom-up | 192 | 0 | 65 | **100** | 74.7 | 85.5 |
| SSD | Single SSD | 166 | 2 | 91 | 98.8 | 64.6 | 78.1 |
| | Top-down | 190 | 0 | 67 | **100** | **73.9** | **85.0** |
| | Bottom-up | 156 | 2 | 101 | 98.7 | 60.7 | 75.2 |
| EfficientDet | Single EfficientDet | 209 | 10 | 48 | 95.4 | 81.3 | 87.8 |
| | Top-down | 211 | 10 | 46 | **95.5** | **82.1** | **88.3** |
| | Bottom-up | 197 | 10 | 60 | 95.2 | 76.7 | 84.9 |
| CenterNet | Single CenterNet | 139 | 7 | 118 | 95.2 | 54.1 | 69.0 |
| | Top-down | 171 | 0 | 86 | **100** | **66.5** | **79.9** |
| | Bottom-up | 135 | 2 | 122 | 98.5 | 52.5 | 68.5 |

Table 11: Performance comparison between the bottom-up, top-down and corresponding single detection model in Video 8.

| **Detector** | **Approach** | **#TP** | **#FP** | **#FN** | **Precision**(%) | **Recall**(%) | **F1**(%) |
|---|---|---|---|---|---|---|---|
| FasterR-CNN | Single Faster R-CNN | 163 | 77 | 233 | 67.9 | 42.2 | 52.1 |
| | Top-down | 201 | 82 | 185 | 71.0 | **52.1** | **60.1** |
| | Bottom-up | 163 | 53 | 223 | **75.5** | 42.2 | 54.2 |
| SSD | Single SSD | 261 | 101 | 125 | 72.1 | **67.6** | 69.8 |
| | Top-down | 247 | 29 | 139 | **89.5** | 64.0 | **74.6** |
| | Bottom-up | 261 | 101 | 125 | 72.1 | **67.6** | 69.8 |
| EfficientDet | Single EfficientDet | 264 | 166 | 122 | 61.4 | 68.4 | 64.7 |
| | Top-down | 333 | 26 | 53 | **92.8** | **86.3** | **89.4** |
| | Bottom-up | 264 | 158 | 122 | 62.6 | 68.4 | 65.3 |
| CenterNet | Single CenterNet | 31 | 146 | 355 | 17.5 | 8.0 | 11.0 |
| | Top-down | 134 | 41 | 252 | **76.6** | **34.7** | **47.8** |
| | Bottom-up | 31 | 58 | 355 | 34.8 | 8.0 | 13.1 |

Table 12: Performance comparison between the bottom-up, top-down and corresponding single detection model in Video 9.

| **Detector** | **Approach** | **#TP** | **#FP** | **#FN** | **Precision**(%) | **Recall**(%) | **F1**(%) |
|---|---|---|---|---|---|---|---|
| FasterR-CNN | Single Faster R-CNN | 132 | 12 | 108 | 91.7 | **55.0** | **68.8** |
| | Top-down | 110 | 10 | 130 | 91.7 | 45.8 | 61.1 |
| | Bottom-up | 132 | 12 | 108 | 91.7 | **55.0** | **68.8** |
| SSD | Single SSD | 214 | 12 | 26 | 94.7 | **89.2** | **91.8** |
| | Top-down | 156 | 17 | 84 | 90.2 | 65.0 | 75.5 |
| | Bottom-up | 206 | 7 | 34 | **96.7** | 85.8 | 90.9 |
| EfficientDet | Single EfficientDet | 228 | 26 | 12 | 89.8 | **95.0** | 92.3 |
| | Top-down | 221 | 2 | 19 | **99.1** | 92.1 | **95.5** |
| | Bottom-up | 221 | 26 | 19 | 89.5 | 92.1 | 90.8 |
| CenterNet | Single CenterNet | 113 | 12 | 127 | 90.4 | 47.1 | 61.9 |
| | Top-down | 151 | 0 | 89 | **100** | **62.9** | **77.2** |
| | Bottom-up | 108 | 10 | 132 | 91.5 | 45.0 | 60.3 |

Table 13: Performance comparison between the bottom-up, top-down and corresponding single detection model in Video 10.

| **Detector** | **Approach** | **#TP** | **#FP** | **#FN** | **Precision**(%) | **Recall**(%) | **F1**(%) |
|---|---|---|---|---|---|---|---|
| FasterR-CNN | Single Faster R-CNN | 77 | 96 | 146 | 44.5 | 34.5 | 38.9 |
| | Top-down | 204 | 38 | 19 | **84.3** | **91.5** | **87.7** |
| | Bottom-up | 77 | 96 | 146 | 44.5 | 34.5 | 38.9 |
| SSD | Single SSD | 58 | 94 | 165 | 38.2 | 26.0 | 30.9 |
| | Top-down | 166 | 41 | 57 | **80.2** | **74.4** | **77.2** |
| | Bottom-up | 58 | 94 | 165 | 38.2 | 26.0 | 30.9 |
| EfficientDet | Single EfficientDet | 115 | 142 | 108 | 44.7 | 51.6 | 47.9 |
| | Top-down | 223 | 17 | 0 | **92.9** | **100** | **96.3** |
| | Bottom-up | 115 | 139 | 108 | 45.3 | 51.6 | 48.2 |
| CenterNet | Single CenterNet | 7 | 24 | 216 | 22.6 | 3.1 | 5.5 |
| | Top-down | 168 | 29 | 55 | **85.3** | **75.3** | **80.0** |
| | Bottom-up | 7 | 19 | 216 | 26.9 | 3.1 | 5.6 |

Table 14: Performance comparison between the bottom-up, top-down and corresponding single detection model in Video 11.

| **Detector** | **Approach** | **#TP** | **#FP** | **#FN** | **Precision**(%) | **Recall**(%) | **F1**(%) |
|---|---|---|---|---|---|---|---|
| FasterR-CNN | Single Faster R-CNN | 355 | 53 | 55 | 87.0 | 86.6 | 86.8 |
| | Top-down | 379 | 142 | 31 | 72.7 | **92.4** | 81.4 |
| | Bottom-up | 355 | 41 | 55 | **89.6** | 86.6 | **88.1** |
| SSD | Single SSD | 259 | 190 | 151 | 57.7 | 63.2 | 60.3 |
| | Top-down | 345 | 24 | 65 | **93.5** | **84.1** | **88.6** |
| | Bottom-up | 259 | 182 | 151 | 58.7 | 63.2 | 60.9 |
| EfficientDet | Single EfficientDet | 319 | 250 | 91 | 56.1 | 77.8 | 65.2 |
| | Top-down | 398 | 2 | 12 | **99.5** | **97.1** | **98.3** |
| | Bottom-up | 319 | 250 | 91 | 56.1 | 77.8 | 65.2 |
| CenterNet | Single CenterNet | 161 | 89 | 249 | 64.4 | 39.3 | 48.8 |
| | Top-down | 326 | 43 | 84 | **88.3** | **79.5** | **83.7** |
| | Bottom-up | 161 | 89 | 249 | 64.4 | 39.3 | 48.8 |

Table 15: Performance comparison between the bottom-up, top-down and corresponding single detection model in Video 12.

| **Detector** | **Approach** | **#TP** | **#FP** | **#FN** | **Precision**(%) | **Recall**(%) | **F1**(%) |
|---|---|---|---|---|---|---|---|
| FasterR-CNN | Single Faster R-CNN | 199 | 48 | 55 | 80.6 | 78.3 | 79.4 |
| | Top-down | 225 | 22 | 29 | **91.1** | **88.6** | **89.8** |
| | Bottom-up | 199 | 48 | 55 | 80.6 | 78.3 | 79.4 |
| SSD | Single SSD | 168 | 204 | 86 | 45.2 | 66.1 | 53.7 |
| | Top-down | 170 | 50 | 84 | **77.3** | **66.9** | **71.7** |
| | Bottom-up | 168 | 204 | 86 | 45.2 | 66.1 | 53.7 |
| EfficientDet | Single EfficientDet | 216 | 96 | 38 | 69.2 | 85.0 | 76.3 |
| | Top-down | 221 | 31 | 33 | **87.7** | **87.0** | **87.4** |
| | Bottom-up | 216 | 96 | 38 | 69.2 | 85.0 | 76.3 |
| CenterNet | Single CenterNet | 17 | 24 | 237 | 41.5 | 6.7 | 11.5 |
| | Top-down | 170 | 17 | 84 | **90.9** | **66.9** | **77.1** |
| | Bottom-up | 17 | 22 | 237 | 43.6 | 6.7 | 11.6 |

Table 16: Performance comparison between the bottom-up, top-down and corresponding single detection model in Video 13.

| **Detector** | **Approach** | **#TP** | **#FP** | **#FN** | **Precision**(%) | **Recall**(%) | **F1**(%) |
|---|---|---|---|---|---|---|---|
| FasterR-CNN | Single Faster R-CNN | 314 | 10 | 53 | **96.9** | 85.6 | **90.9** |
| | Top-down | 324 | 55 | 43 | 85.5 | **88.3** | 86.9 |
| | Bottom-up | 314 | 10 | 53 | **96.9** | 85.6 | **90.9** |
| SSD | Single SSD | 168 | 53 | 199 | 76.0 | 45.8 | 57.1 |
| | Top-down | 288 | 17 | 79 | **94.4** | **78.5** | **85.7** |
| | Bottom-up | 168 | 50 | 199 | 77.1 | 45.8 | 57.4 |
| EfficientDet | Single EfficientDet | 298 | 103 | 69 | 74.3 | 81.2 | 77.6 |
| | Top-down | 338 | 7 | 29 | **98.0** | **92.1** | **94.9** |
| | Bottom-up | 298 | 103 | 69 | 74.3 | 81.2 | 77.6 |
| CenterNet | Single CenterNet | 53 | 38 | 314 | 58.2 | 14.4 | 23.1 |
| | Top-down | 214 | 0 | 153 | **100** | **58.3** | **73.7** |
| | Bottom-up | 53 | 26 | 314 | 67.1 | 14.4 | 23.8 |

Table 17: Performance comparison between the bottom-up, top-down and corresponding single detection model in Video 14.

| **Detector** | **Approach** | **#TP** | **#FP** | **#FN** | **Precision**(%) | **Recall**(%) | **F1**(%) |
|---|---|---|---|---|---|---|---|
| FasterR-CNN | Single Faster R-CNN | 415 | 53 | 60 | 88.7 | 87.4 | 88.0 |
| | Top-down | 418 | 50 | 57 | 89.3 | **88.0** | 88.7 |
| | Bottom-up | 415 | 38 | 60 | **91.6** | 87.4 | **89.4** |
| SSD | Single SSD | 370 | 278 | 105 | 57.1 | 77.9 | 65.9 |
| | Top-down | 446 | 14 | 29 | **97.0** | **93.9** | **95.4** |
| | Bottom-up | 370 | 278 | 105 | 57.1 | 77.9 | 65.9 |
| EfficientDet | Single EfficientDet | 434 | 259 | 41 | 62.6 | 91.4 | 74.3 |
| | Top-down | 461 | 24 | 14 | **95.1** | **97.1** | **96.0** |
| | Bottom-up | 434 | 259 | 41 | 62.6 | 91.4 | 74.3 |
| CenterNet | Single CenterNet | 43 | 53 | 432 | 44.8 | 9.1 | 15.1 |
| | Top-down | 391 | 0 | 84 | **100** | **82.3** | **90.3** |
| | Bottom-up | 43 | 48 | 432 | 47.3 | 9.1 | 15.2 |

Table 18: Performance comparison between the bottom-up, top-down and corresponding single detection model in Video 15.

| **Detector** | **Approach** | **#TP** | **#FP** | **#FN** | **Precision**(%) | **Recall**(%) | **F1**(%) |
|---|---|---|---|---|---|---|---|
| FasterR-CNN | Single Faster R-CNN | 204 | 182 | 60 | 76.6 | 52.8 | 62.5 |
| | Top-down | 281 | 74 | 106 | **79.1** | **72.7** | **75.7** |
| | Bottom-up | 204 | 55 | 182 | 78.7 | 52.8 | 63.2 |
| SSD | Single SSD | 17 | 94 | 370 | 15.2 | 4.3 | 6.8 |
| | Top-down | 319 | 14 | 67 | **95.7** | **82.6** | **88.7** |
| | Bottom-up | 17 | 94 | 370 | 15.2 | 4.3 | 6.8 |
| EfficientDet | Single EfficientDet | 72 | 178 | 314 | 28.8 | 18.6 | 22.6 |
| | Top-down | 322 | 60 | 65 | **84.3** | **83.2** | **83.8** |
| | Bottom-up | 72 | 173 | 314 | 29.4 | 18.6 | 22.8 |
| CenterNet | Single CenterNet | 7 | 17 | 379 | 30 | 1.9 | 3.5 |
| | Top-down | 286 | 26 | 101 | **91.5** | **73.9** | **81.8** |
| | Bottom-up | 5 | 5 | 382 | 50 | 1.2 | 2.4 |

Table 19: Performance comparison between the bottom-up, top-down and corresponding single detection model in Video 16.

| **Video** | **Approach** | **Precision**(%) | **Recall**(%) | **F1**(%) |
|---|---|---|---|---|
| Video 1 - Building entry door outdoor | Single detector | 93.3 | 70.0 | 80.0 |
| 415 frames | Top-down | 92.0 | **80.8** | **86.0** |
| | Bottom-up | **95.8** | 69.5 | 80.6 |
| Video 2 - Building entry door outdoor | Single detector | 92.6 | 71.1 | 80.4 |
| 545 frames | Top-down | 90.9 | **82.4** | **86.4** |
| | Bottom-up | **95.0** | 71.1 | 81.3 |
| Video 3 - Building entry door outdoor | Single detector | 69.6 | 32.8 | 44.6 |
| 293 frames | Top-down | **94.1** | **94.5** | **94.3** |
| | Bottom-up | 69.7 | 32.8 | 44.7 |
| Video 4 - Building entry door outdoor | Single detector | 62.7 | 54.0 | 58.0 |
| 276 frames | Top-down | **88.0** | **90.8** | **89.3** |
| | Bottom-up | 62.7 | 54.0 | 58.0 |
| Video 5 - Back garage door outdoor | Single detector | 79.8 | 48.2 | 60.1 |
| 319 frames | Top-down | **92.2** | **78.0** | **84.5** |
| | Bottom-up | 82.3 | 45.8 | 58.9 |
| Video 6 - Back garage door outdoor | Single detector | 57.6 | 59.1 | 58.3 |
| 240 frames | Top-down | **97.0** | **97.8** | **97.4** |
| | Bottom-up | 61.1 | 59.1 | 60.1 |
| Video 7 - Back garage door outdoor | Single detector | 97.8 | 75.9 | 85.5 |
| 324 frames | Top-down | 96.8 | **83.5** | **89.7** |
| | Bottom-up | **98.7** | 73.1 | 84.0 |
| Video 8 - Back garage door outdoor | Single detector | 97.4 | 69.4 | 81.0 |
| 257 frames | Top-down | **98.6** | **76.0** | **85.8** |
| | Bottom-up | 98.1 | 66.1 | 79.0 |
| Video 9 - Service desk indoor | Single detector | 54.7 | 46.6 | 50.3 |
| 386 frames | Top-down | **82.5** | **59.3** | **69.0** |
| | Bottom-up | 61.2 | 46.6 | 52.9 |
| Video 10 - Service desk indoor | Single detector | 91.6 | **71.6** | **80.4** |
| 240 frames | Top-down | **95.2** | 66.5 | 78.3 |
| | Bottom-up | 92.3 | 69.5 | 79.3 |
| Video 11 - Transit area indoor | Single detector | 37.5 | 28.8 | 32.6 |
| 223 frames | Top-down | **85.7** | **85.3** | **85.5** |
| | Bottom-up | 38.7 | 28.8 | 33.0 |
| Video 12 - Transit area indoor | Single detector | 66.3 | 66.7 | 66.5 |
| 410 frames | Top-down | **88.5** | **88.3** | **88.4** |
| | Bottom-up | 67.2 | 66.7 | 67.0 |
| Video 13 - Transit area indoor | Single detector | 59.1 | 59.1 | 59.1 |
| 254 frames | Top-down | **86.7** | **77.4** | **81.8** |
| | Bottom-up | 59.6 | 59.1 | 59.3 |
| Video 14 - Transit area indoor | Single detector | 76.4 | 56.7 | 65.1 |
| 367 frames | Top-down | **94.5** | **79.3** | **86.2** |
| | Bottom-up | 78.8 | 56.7 | 66.0 |
| Video 15 - Transit area indoor | Single detector | 63.3 | 66.4 | 64.8 |
| 475 frames | Top-down | **95.3** | **90.3** | **92.8** |
| | Bottom-up | 64.6 | 66.4 | 65.5 |

Table 20: Performance comparison between the bottom-up, top-down and corresponding single detection model over the fifteen test videos. The result for each pair (video, approach) is calculated using the four considered single detection models.