Automatic Detection of Knives in Complex Scenes

Maira Moran, Aura Conci and Ángel Sánchez

Abstract Smart Cities use a variety of Information and Communication Technologies (ICT) and databases to improve the efficiency and efficacy of city services. Security is one of the main topics of interest in this context. The increase in crime rates demands the development of new solutions for detecting possible violent situations. Video surveillance (CCTV) cameras can provide a large amount of valuable information contained in images which can be difficult to be analyzed by humans in an efficient form. Identifying and classifying weapons in such images is a challenging problem that can be driven by the application of Deep Learning techniques. Object detection algorithms, especially advanced Machine Learning ones, have demonstrated impressive results in a wide range of applications. However, they can fail in certain application scenarios. This work describes a novel proposal for knife detection in complex images. This is a challenging problem due to the multiple variabilities of these objects in scenes (i.e., changing shapes, sizes and illumination conditions, among others), which can negatively impact the performance of mentioned algorithms. Our approach analyzed the combination two super-resolution techniques (as a preprocessing stage) with one object detection network to effectively solve the considered problem. The results of our experiments show that the proposed methodology can produce better results when detecting small objects having reflecting surfaces (i.e., knives) in scenes. Moreover, the approach could be adapted for surveillance applications that need real-time detection of knives in places monitored by cameras.

Maira Moran
IC - UFF, 24210-310 Niterói (Rio de Janeiro), Brazil, e-mail: mhernandez@id.uff.br

Aura Conci
IC - UFF, 24210-310 Niterói (Rio de Janeiro), Brazil, e-mail: aconci@ic.uff.br

Ángel Sánchez
ETSII - URJC, 28933 Móstoles (Madrid), Spain, e-mail: angel.sanchez@urjc.es

1 Introduction

New Smart City (SC) technologies are helping cities to maximize their resources and increase efficiencies in all facets of urban life. A SC consists of an urban space where Information and Communication Technologies (ICT) are extensively used to improve the quality and performance of services such as transportation, energy, water or infrastructures, in order to reduce resource energy consumption, wastage and overall costs [11].

One of the relevant areas in the SC is guarantying the security of their citizens. Video surveillance CCTV cameras, which are commonly used by urban police departments, can be part of these "smart" technologies in combination with video analytics software. Video recordings contain a wealth of valuable information that can be automatically analyzed to detect anomalous (and even dangerous) events from multiple cameras. Commonly, in security centers work human operators that are in charge of a large number of CCTV cameras, capturing multiple city views, operating in real-time. Due to the difficulty of humans for being able to keep their attention during several hours in front of many cameras (usually, more than 16), it is desirable that the video surveillance system could be automatically able to recognize potentially critical security events in specific video frames and cameras. In such cases, the system can notify an alert to the human operators to focus his/her attention on a concrete camera. Image-content analytics technology can help solving the event detection problem by processing video frames and identifying, classifying and indexing some types of targets objects (e.g., cars, motorcycles, persons or animals) [19]. Driven by Artificial Intelligence techniques, surveillance software can also make these images (or frames) in videos as searchable, actionable and quantifiable.

In this context, this work presents a study of applying deep networks to the problem of automatically detecting knives (and related objects) in images. This is a challenging problem due to the multiple variabilities of these targets when appearing in scenes. In particular, the changing shapes of knives, their relatively small sizes in images, the possibility of being partially occluded, being carried by a person (or being free) in a location, the changing illumination conditions in scenes, among other difficulties. All these involved variabilities (which can also appear combined), can produce a negative impact over the performance of the detection algorithms. The extension of this work to detect firearms like guns would not be difficult, since the used models are configurable for including additional object classes.

This paper describes a research on the application of combining super-resolution techniques with deep neural networks to effectively handle the knife detection problem in complex images. Our results show that the proposed methodology produces accurate results when detecting this special type of objects. It organized as follows. Section 2 summarizes the related work on the considered knife detection problem. The aspects of small-object detection (and, in particular, knives), as well as the description of the YOLOv4 model used in this work, are described in Section 3. Sections 4 and 5 describe, respectively, the dataset used in the experiment and some related pre-processing on it. The experiments carried out and their analysis appear in Section 6. Finally, Section 7 concludes this work.

2 Related work

The problem of small-sized object detection in labeled datasets is still not solved at all [16]. In this problem, very few image pixels represent the whole objects of interest, which make it difficult to detect and classify them. The use of super-resolution to increase the object size in order to compensate for the loss of object information can help to the detection task [17].

One specific use case of small-sized object detection consist in the detection of knives. As for other types of weapons, carrying knives in public is either forbidden or restricted in many countries. Since knives are both widely available and can be used as weapons, their detection is of high importance for security personnel [8].

One of the first works on automatic detection of knives was presented by Kmiec and Glowacz in 2011 [12]. These authors compute a set of image descriptors using Histograms of Oriented Gradients (HOG). These descriptors, that are invariant to geometric and photometric transformations, are used with a SVM for the detection task.

Glowacz and collaborators [8] propose an Active Appearance Model (AAM) to detect knives in images. As the knife-blade has usually an uniform texture, using an AAM could contribute to improve detections, since the model would not converge to other objects having a similar shape.

In 2016 Grega et al.[10] publish a highly-cited work on detection of firearms and knives from CCTV images. Their goal is to reduce the number of false alarms in detections. These authors use a modified sliding window technique to determine the approximate position of the knife in an image. Then, they extract edge histograms and texture descriptors to create feature vectors for training a SVM able to classify the detected objects as knives.

Buckchash and Raman [2] have proposed in 2017 a method to detect visual knives in images. Their approach has three stages: foreground segmentation, feature extraction using the FAST (Feature Accelerated Segment Test) corner detector, and Multi-Resolution Analysis (MRA) for classification and target confirmation.

More recent works make use of deep networks. Castillo et al. [3] presented a system to locate cold steel weapons in images. (such as knives). These weapons have a reflecting surface that under different light conditions can distort and/or blur their shape in the frames. To solve the problem, the authors propose the combination of a contrast-enhancement brightness-guided preprocessing procedure with the use of different types of Convolutional Neural Networks (CNN).

Other authors have experimented with infrarred images (IR) to detect not visible (i.e., hidden) knives [18]. A type of deep neural network (GoogleNet), that was trained on natural images, was fine-tuned to classify the IR images as people or as people carrying a hidden knife.

A very comprehensive survey on the progress of Computer Vision-based concepts, methodologies, analysis and applications for automatic knife detection has been published recently showing the state-of-the-art of vision-based detection systems [4]. The authors define a taxonomy based on the state-of-the-art methods for knife detection. They analyzed several image features used in the considered works for

this task. The challenges regarding weapon detection and new-frontier in weapon detection are included, as well. This survey references more then 80 works, and concludes pointing out some possible research gaps in the problem and related ones.

Another brief review of the state-of-the-art approaches of knife identification and classification was published very recently [5]. Although, this article is not a review paper, it presents a broad analysis of recent works using Convolutional Neural Network (CNN), Recurrent Convolutional Neural Network (R-CNN), Faster R-CNN, and Overfeat Network, that is most of deep learning methods used up now for the considered problem.

3 YOLOv4 architecture for detection of knives

This section summarizes the object detection problem particularized for the case of knives, and the features of YOLOv4 model used in our experiments.

3.1 Detection of knives

Object detection is a challenging task in Computer Vision that has received large attention in last years, especially with the development of Deep Learning [19] [16]. It presents many applications related with video surveillance, automated vehicle system robot vision or machine inspection, among many others. The problem consists in recognizing and localizing some classes of objects present in a static image or in a video. Recognizing (or classifying) means determining the categories (from a given set of classes) of all object instances present in the scene together with their respective network confidence values on these detections. Localizing consists in returning the coordinates of each bounding box containing any considered object instance in the scene. The detection problem is different from (semantic) instance segmentation where the goal is identifying for each pixel of the image the object instance (for every considered type of object) to which the pixel belongs. Some difficulties in the object detection problem include aspects such as geometrical variations like scale changes (e.g., small size ratio between the object and the image containing it) and rotations of the objects (e.g., due to scene perspective the objects may not appear as frontal); partial occlusion of objects by other elements in the scene; illumination conditions (i.e., changes due to weather conditions, natural or artificial light); among others but not limited to these ones. Note that some images may contain several combined variabilities (e.g., small, rotated and partially occluded objects). In addition to detection accuracy, another important aspect to consider is how to speed up the detection task.

Detecting knives in images (and also in videos) is a challenging problem. The images where these objects can present several extrinsic and intrinsic variabilities due to the size of the target object (in general, its size ratio is very small when

compared to the image size), the possibility of the weapon being carried by a person or appearing freely placed in a location, the illumination conditions of the scene (which could produce a very low contrast between the knife and the surrounding background), among other real difficulties.

3.2 YOLOv4

Redmon and collaborators have proposed in 2016 the new object detector model called YOLO (acronym of "You Only Look Once") [15], which handles the object detection as a one-stage regression problem by taking an input image and learning simultaneously the class probabilities and the bounding box object coordinates. This first version of YOLO was also called YOLOv1, and since them the successive improved versions of this architecture (YOLOv2, YOLOv3, YOLOv4, and YOLOv5, respectively) have gained much popularity within the Computer Vision community.

Different from previous two-stage detection networks, like R-CNN and faster R-CNN, the YOLO model used only one-stage detection. That is, it can make predictions with only one "pass" in the network. This feature made the YOLO architecture extremely fast, at least 1000 times faster than R-CNN and 100 times faster than Fast R-CNN.

The architecture of all YOLO models have some similar components which are summarized next:

- *Backbone*: A convolutional neural network that produces and accumulates visual features with different shapes and sizes. Classification models like ResNet, VGG, and EfficientNet are used as feature extractors.
- *Neck*: This component consists in a set of layers that receive the output features extracted by the Backbone (at different resolutions), and integrate and blend these characteristics before passing them on to the prediction layer. For example, models like Feature Pyramid Networks (FPN) or Path Aggregation networks (PAN) have been used for such purpose.
- *Head*: This component takes in features from the Neck along with the bounding box predictions. It performs the classification along with regression on the features and produces the bounding box coordinates to complete the detection process. Generally, it produces four output values per detection: the x and y center coordinates, and width and height of detected object, respectively.

Next, we summarize the main specific features of YOLOv4 architecture that were used in our experiments. YOLOV4 was released by Alexey Bochkovskiy et al. in their 2020 paper “YOLOv4: Optimal Speed and Accuracy of Object Detection” [1]. This model is ahead in performance on other convolutional detection models like EfficientNet and ResNext50. Like YOLOv3, it has the Darknet53 model as Backbone component. It has a speed of 62 frames per second with an mAP of 43.5% on the MS COCO dataset.

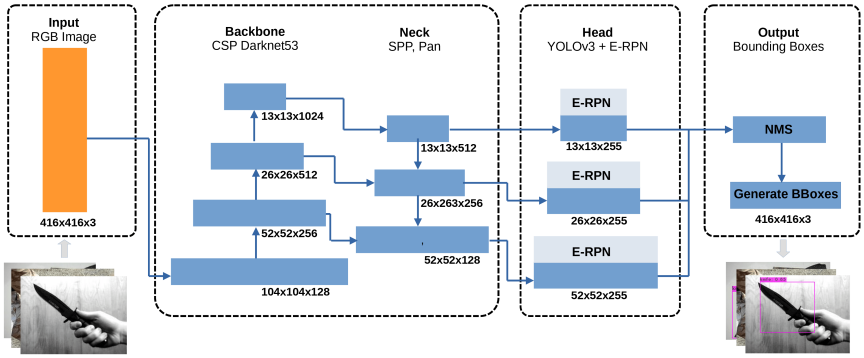


Fig. 1: Schematic representation of YOLOv4 architecture.

As technical improvements with respect to YOLOv3, YOLOv4 introduces as new elements the bag of freebies and the bag of specials.

Bag of Freebies (BoF) are a set of techniques enabling an improvement of the model in performance without increasing the inference cost. In particular:

- *Data augmentation techniques*: CutMix, MixUp, CutOut, ...
- *Bounding box regression loss types*: MSE, IoU, CIoU, DIoU, ...
- *Regularization techniques*: Dropout, DropPath, DropBlock, ...
- *Normalization techniques*: Mini-batch, Iteration-batch, GPU normalization, ...

Bag of Specials (BoS) consist in techniques that increase accuracy while slightly increasing the computation cost. In particular:

- *Spatial Attention Modules* (SAM): Spatial Attention (SA), Channel-wise Attention (CA), ...
- *Non-Max Suppression modules* (NMS)
- *Non-linear activation functions*: ReLU, SELU, Leaky, Mish, ...
- *Skip-Connections*: Weighted Residual Connections(WRC), Cross-Stage Partial connections (CSP), ...

Figure 1 illustrates the layer structure of YOLOv4 network used in our experiments.

4 Datasets

The success of the proposed method is highly related to the quality of the data used to train the supervised algorithm. One of the main applications for the proposed problem is its inclusion in surveillance system. To our knowledge there are no current publicly-available CCTV datasets. The datasets used in similar works consist of images captured by the authors, and many of them are taken from the Internet. In

this section, we present two main datasets in this field, which are also used in our work for training and testing the models.

4.1 DaSCI dataset

The DaSCI knives dataset [14] is a subset of a more general weapon detection dataset. It was created by people from University of Granada as an open data repository, and designed for the object detection task. The annotation files describe the image region where each knife is located, by defining a correspondent bounding box. It is composed of 2,078 images, each one of them containing at least one knife, resulting a total of 2,155 objects. The dataset was ,created considering the diversity of the objects (i.e., the images were selected in order to provide samples with different visual features), resulting in a robust challenge dataset. Some considered visual features of knives are: types, shapes, colors, sizes, materials, locations, positions in relation to other scene objects, indoor/outdoor scenarios, and so on. The images were extracted mostly from the Internet, and the main sources were free image stocks and YouTube videos, from which frames were extracted, considering the criteria previously mentioned. The dataset is divided into 15 subsets (referred as DS1-DS15) according with their image sources. Each one is composed by: 8, 130, 16, 12, 188, 242, 11, 36, 49, 130, 603, 29, 143, 108, and 83 images, respectively. Table 1 summarizes the information about these subsets. Figure 2 shows some examples of images extracted from some of these sources.

Source type	Video frames	DS1, DS2, DS3, DS4, DS5, DS6, DS7, DS8, DS9, DS12, DS13, DS14, DS15
	Internet images	DS11
	Captured by authors	DS10
Objects per image	One	DS1, DS2, DS3, DS4, DS5, DS6, DS7, DS8
	Multiple	DS9, DS10, DS11, DS12, DS13, DS14, DS15
Multiple scenarios	Yes	DS1, DS2, DS3, DS4, DS5, DS6, DS7, DS8, DS9
	No	DS10, DS11, DS12, DS13, DS14, DS15

Table 1: Information in DaSCI subsets

As previously mentioned, the size, position and location if the objects varies in the dataset. This way, the area that the each knife covers in the image also differs (although it is often very small). Figure 3 shows histograms of these proportions. Even considering that the dataset was designed to present a high heterogeneity in this aspect, it can be observed that many of the objects (i.e., around 50%) only cover between 1% and 20% of the image size. The remaining objects are more equally distributed, occupying different portions of their respective images.

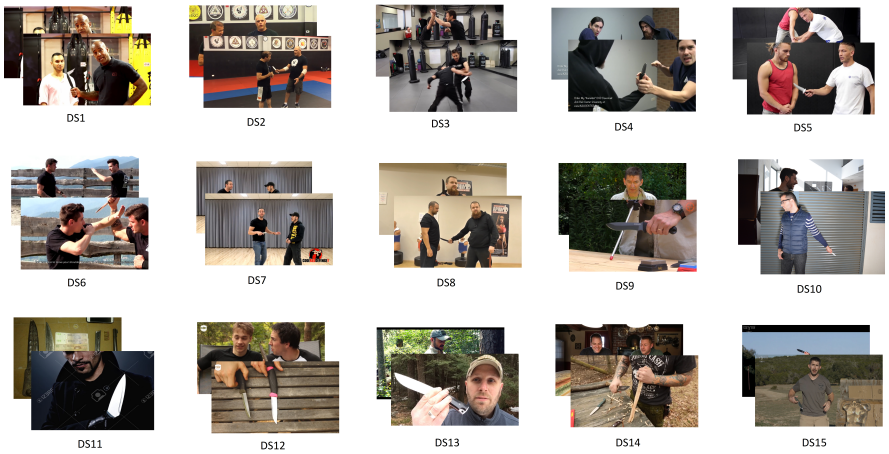
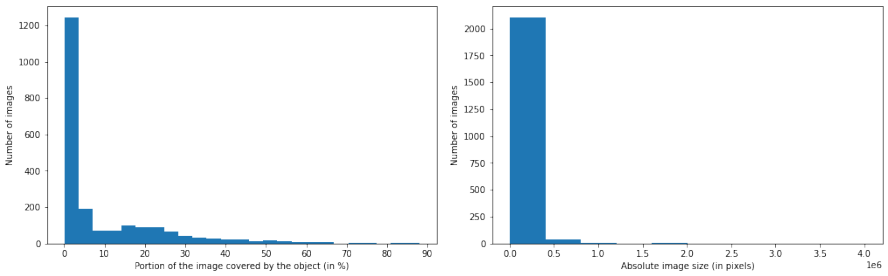


Fig. 2: Samples of each DaSCI subset

The fact that knives in this dataset tend to occupy a small area over the images (and consequently, present a low spatial resolution) is a challenging issue for the detection task, that can be assessed in the pipeline of possible solutions to be developed.



(a) (b)

Fig. 3: Histogram of object sizes composing knives samples in DaSCI dataset: (a) relative object vs image size proportions and (b) absolute size (spatial resolution).

It is important to mention that the annotations are not completely uniform, in the sense that for some cases the knife area described in the annotation file covers the whole knife (i.e., both blade and handle), and for other cases the described knife are cover only the knife blade.

The annotation formats describe each image and the positions of the associated objects. Firstly, the image information is detailed, including its file name, path and dimensions (width, height and depth, being this last one related to the number of channels, mostly 3 since RGB color images are used). Then, the information of

Fig. 4: Example of images composing the DaSCI dataset and respective annotations.

objects is listed (always 'knife' in this work), and its respective region, which is described as a bounding box denoted by coordinates of its top left (x_{min},y_{min}) and bottom right (x_{max}, y_{max}) corners.

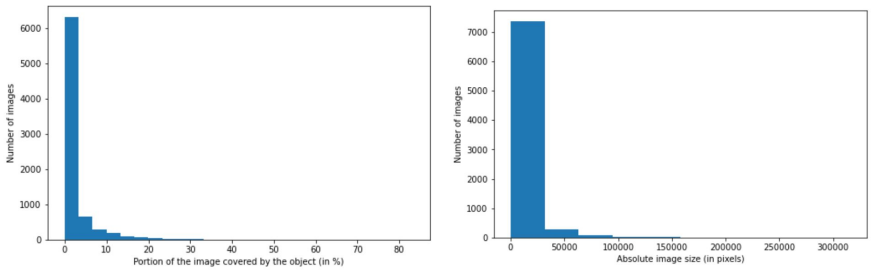
4.2 MS COCO dataset

The MS COCO (Microsoft Common Objects in Context) dataset [6] is widely used in Computer Vision literature for object detection and segmentation tasks. Since the appearance of its first version, other upgraded versions from this dataset have been published. In this work, we consider the MS COCO 2017 dataset. It consists of a very large and complete dataset, composed of 330,000 images with 1.5 million objects. This dataset has 80 different classes, and class 'knife' is one of them with 7,770 labeled objects from 4,326 images. Since the MS COCO dataset was initially designed to encompass objects of 80 different classes, the images selected to compose it mostly portrait scenes crowded with different objects, and knives are not the main object of interest in the scene. This can also be considered as a challenging issue for the problem assessed in this study. Figure 5 shows some samples of the MS COCO dataset.

Fig. 5: Example of images composing COCO dataset and respective annotations.

Also, as similarly to DaSCI, in this dataset the knives mainly present a very low spatial resolution, which is another aspect to be handled in this study. Figure 6 shows an histogram of the object area vs image area ratio for the knives samples.

The object bounding boxes in MS COCO annotations are described by the x and y coordinates of the top left corner, and the object's width and height, respectively.



(a) (b)

Fig. 6: Histogram of sizes for the knives samples in MS COCO dataset: (a) relative object vs image size proportions and (b) absolute size (spatial resolution).

4.3 Knife classification datasets

The knife detection task have been previously assessed in the literature (see survey work [4]). However, the number of public datasets available is still very limited. Regarding datasets that include knives in images, there are some available options that were initially proposed for classification tasks. Although their annotation should be expanded in order to be employed in a detection task, it is important to consider that such datasets are also available.

There is another dataset provided by DaSCI that could be employed for the knife classification task, composed of 10,039 images, which were extracted from the Internet. The annotations cover 100 object classes, being 'knife' the target one, with 635 images. Among the others classes included are: 'car', 'plant', 'pen', 'smartphone', 'cigar', etc.

Grega et. al. [9] also proposed a method for knife classification. Their dataset consists of 12,899 images at 100×100 resolution, from which 9,340 are negative samples, and 3,559 are positive ones. The positive samples consist of a scene with a knife held in a hand, and the negative samples consists of scenes with no knife. Concerning the environment, the scenes in the images can be indoor and outdoor.

5 Pre-processings on dataset

5.1 Dataset preparation

In the YOLOv4 model each annotated file presents the following structure: object class, object coordinates (x and y), $width$ and $height$, separated by a simple space:

```
0 x y width height
```

In a YOLOv4 annotation file, each line corresponds to an object. An example of annotation in this format is shown next:

```
0 25 40 100 120
0 30 15 80 50
```

Note that each annotation file refers to an image, that contains one or more objects. In the example above, the first line describes the first object, that is the object class 'knife' (denoted by '0'). Also, the upper left corner of this first object's bounding box is in the position $x = 25$ and $y = 40$. Finally, this first object has a width of 100 and a height of 120. Similarly, for the second object in the example annotation.

As previously mentioned, the object regions in the DaSCI annotations are described as bounding boxes defined by the coordinates of the top left (x_{min}, y_{min}) and bottom right (x_{max}, y_{max}) corners. In this way, the values to compose these annotations can be easily calculated from the DaSCI annotations:

```
x = xmin
y = ymin
width = xmax-xmin
height = ymax-ymin
```

This way, YOLOv4 annotation obtained from the DaSCI XML annotation is composed of:

```
0 xmin ymin xmax-xmin ymax-ymin
```

As described in Section 4, the object's bounding box in the MS COCO annotation is also defined by the x and y coordinates of the upper left corner, and the object's width and height, so as in the YOLOv4 annotation format. The information to compose the annotations are directly transcribed from the MS COCO to a JSON annotation file. Note that, in this structure, each object annotation refers to an object, not to an image.

5.1.1 Image pre-processing

The images to be used as input of the YOLOv4 algorithm must present the a spatial resolution of 416×416. In this sense, the images of both MS COCO and DaSCI datasets must be resized to meet this condition. As previously mentioned, both datasets are composed by images with different sizes (i.e., spatial resolutions), so for some images the re-scale would result in an decrease of the image size, and for others this resizing would enlarge the original images. Increasing the image size, can be specially critic, since the methods commonly used for this task consist of interpolations that frequently lead to effects like blur, aliasing, etc., degrading the quality of the resulting image.

In order to observe the impact of the resizing part of the preprocessing, two alternative resizing operations were performed. The first one is bilinear interpolation,

commonly used as a "black box" operation in most machine learning libraries, including the PyTorch Python library used in this work. The second one is SRGAN (Generative Adversarial Network for single image Super-Resolution) [13], which consists of a machine learning supervised algorithm. The SRGAN, more specifically one of its variations, is currently state of the art for some widely known challenges. Considering that the SRGAN uses a generative network G to create high-resolution images which are so similar to the original ones, that can mislead the differentiable discriminator D, which is trained to distinguish between the generated and the real super-resolution image. In this process, the D network demands an evolution of G during the training process, leading to perceptually superior solutions [13]. In this work, the SRGAN training was performed using the ImageNet dataset

On the other hand, the bilinear interpolation calculates the values of the new interpolated points based on a weighted mean of their surrounding points (four neighbors) in the original image. The weight assigned to each neighbor point is based on its distance to the new point. Consequently, the value of the new point is mostly influenced by the values of closer neighbors.

In this experiment, we analyze the impact of using super-resolution as a pre-processing step of the object detection algorithm. For such purpose, we have adopted a cross-dataset evaluation approach. Evaluations configured in an in-domain setting, which is defined by using the samples from the same dataset for training and testing the algorithms, tend to bias and affects negatively the generalization of machine learning algorithms. Moreover, the transfer learning technique was also assessed, as described in section 5.3.

5.2 Dataset variabilities

As previously mentioned, several factors can affect the performance of the proposed algorithms, as the illumination conditions, object size, perspective, visibility, etc. In this sense, we created subsets of interest from the original test set. Each of these test sets presents an special condition, so one can observe how a particular condition affects to the results of the models. Next, the subsets are listed next:

1. Outdoor: it covers all the images that denote outdoors scenes, related mostly to a higher luminosity.
2. Indoor: composed of images that denote indoor scenes, mostly presenting a lower luminosity.
3. Occluded: composed of images in which the knives are being handled by a person, remaining partially occluded.
4. Not occluded: the object is lying on a surface and it is not held by anyone.

These subsets are not exclusive (i.e., the same image can belong to more than one subset), except when the conditions where defined subsets are excluding (e.g., subsets 1 and 2).

Also, the ratio between object size and image size is a factor that can affect the models' performance, specially considering that the use of super-resolution as pre-processing step may influence the results for small objects. As presented in the histogram of Section 4 (see Figure 2), most of the objects that compose the DaSCI database, which is used as test set in our experiments, cover less than 20% of the corresponding images.

5.3 Transfer learning

Along with the previously mentioned super-resolution pre-processing, another technique employed and analyzed in the performed experiments is transfer learning.

The transfer learning applied in this work consisted basically of using weights obtained from a task in a different domain to initialize the object detection algorithm before performing the actual training using the samples of the actual domain (in order to promote a faster convergence of the model). In this work, the initialization of weights was carried out by training a YOLOv4 algorithm using the Pascal VOC dataset. Until the 105-th convolutional layer, the weights obtained by the transfer learning were used, and the remaining layers were re-trained using our final task.

The PASCAL VOC dataset [7] is widely used for supervised tasks such a as classification, detection and segmentation, being employed in benchmark comparisons for such tasks. It is composed of a wide range of images in realistic scenes. Their annotation associate them with twenty different classes. The class 'knife' is not present in this dataset. Three subsets compose it: train, validation and test. The first subset (train) is composed of 1.464 images, the validation set is composed of 1,449 images, and the test set consists of a private set. Figure 7 shows some examples of images from the PASCAL VOC dataset. Even considering that there are other image datasets widely known in literature, such as the ImageNet dataset, we decided to use the PASCAL VOC dataset since their annotations include bounding boxes designed for a detection task.

6 Experimental results

In this section, we present the results of performed experiments, which analyze the impact of using transfer learning and super-resolution techniques in the training process of the object detection network (YOLOv4). In Subsection 6.1, we summarize the metrics used in this analysis. Then, Subsection 6.2 presents the results of experiments, comparing the results obtained by using each of the mentioned techniques, in general and associated with different aspects of the test dataset such as object size, visibility and illumination.

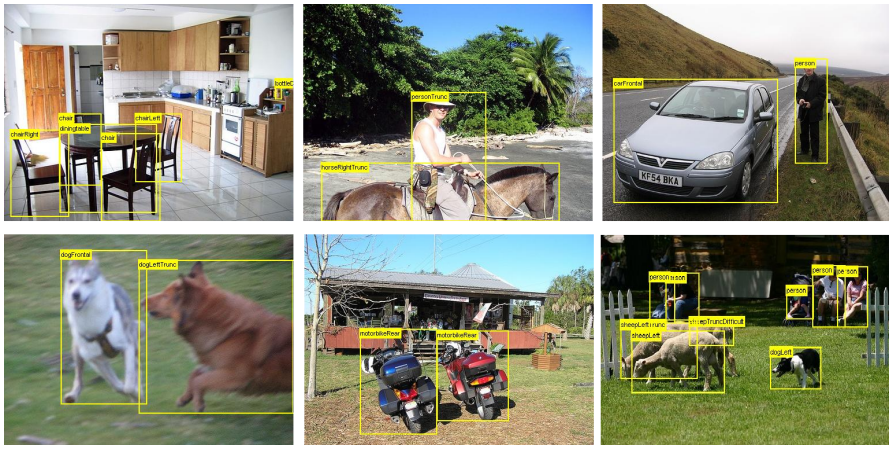


Fig. 7: Example of images that composed the PASCAL VOC dataset.

6.1 Description of Performance Metrics

The evaluation is based on true positives TP (i.e., regions correctly detected as regions containing knives); false negatives FN (i.e., non detected regions containing knives); and false positives FP (i.e., regions incorrectly detected as regions containing knives). From these results some metrics can calculated such as Precision (Prec), Recall (Rec), and F1-Score, using the following equations:

$$Prec = \frac{TP}{TP + FP} \qquad Rec = \frac{TP}{TP + FN} \qquad F1 = 2\frac{Prec \times Rec}{Prec + Rec} \tag{1}$$

The Jaccard index or Intersection over Union (IoU) is also used in this analysis. This metric computes the areas of the bounding boxes denoting the detected knives and the corresponding ground truths.

Figure 8 exemplifies the mentioned IoU areas for several test images. The area in blue represents the bounding box by obtained by one of the proposed algorithms, and the area in violet shows the bounding box defined by the ground truth.

6.2 Experimental Results

Next, we compare the results of different YOLOv4 models trained using the considered approaches. These training models are characterized as shown in Table 2.

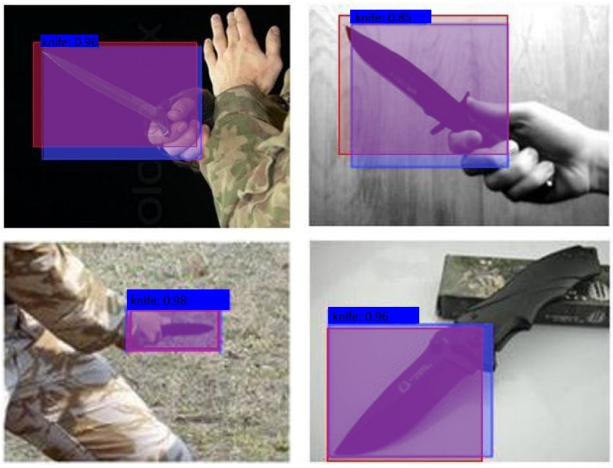


Fig. 8: Bounding box examples: ground truth (violet) and algorithm result (blue).

Table 2: Training variations

Model	Training process	
	Tranfer Learning	Pre-processing
M1	No	Bilinear interpolation
M2	Yes	Bilinear interpolation
M3	No	SRGAN
M4	Yes	SRGAN

6.3 General results

As described in Subsection 5.1.1, we used the cross-dataset approach to train and test all the models. The test dataset (DaSCI) is composed of 2,078 images, which cover 2,155 objects. The main hyper-parameters used in the training process are: confidence prediction threshold = 0.25, IoU threshold = 0.5, and batch size = 1.

Table 3 shows the values obtained for the selected metrics considering the whole dataset. It is possible that not all models detected most of the objects. The best overall performance was achieved by M3. One can notice that the use of transfer learning promotes a worse overall performance for models M2 and M4, compared with M1 and M3. Also, the results suggest that using the super-resolution pre-processing affects the models performance in different ways depending on whether it is combined with transfer learning or not.

For the models not trained with transfer learning (M1 and M3), the SRGAN subtly improved the results, increasing the number of TP in 7 cases and reducing the number of FN in 7 cases. The number of FP was substantially reduced (-111 cases). On the other hand, for the models trained with transfer learning (M2 and M4), the

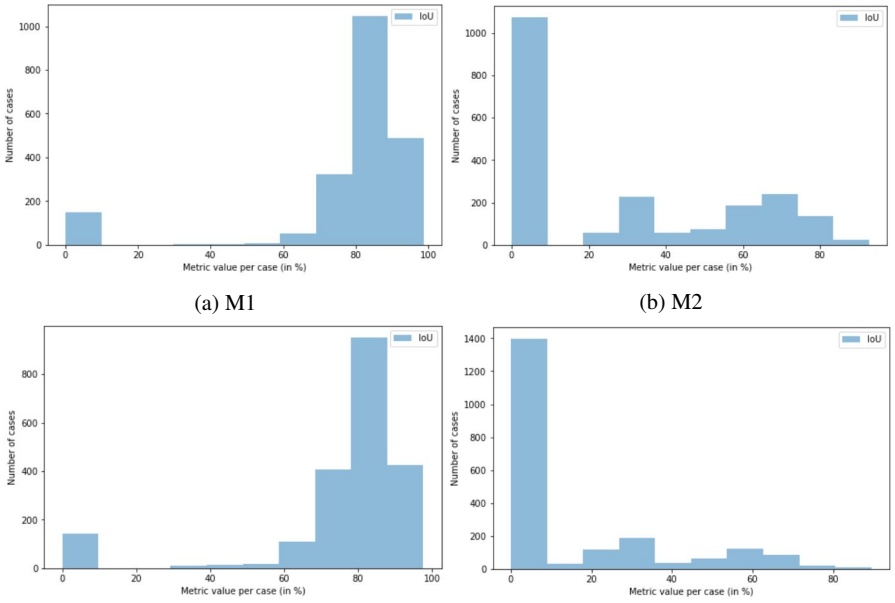
Table 3: General results of the models

Model	TP	FP	FN	IoU (mean)	Precision	Recall	F1-Score
M1	2,057	143	98	0.776	0.935	0.955	0.945
M2	1,071	774	1,084	0.269	0.580	0.497	0.535
M3	2,064	32	91	0.756	0.985	0.958	0.971
M4	727	1,211	1,428	0.141	0.375	0.337	0.355

results using SRGAN were substantially worse. This difference is of -344 (-32.12%) for TP, +437 (56.46%) for FP, and +344 (31.73%) for FN.

Concerning the other performance metrics, the M1 model presented the best average IoU values, and the M3 model presented the best Precision, Recall and F1-score values. In general, the use of the super-resolution pre-processing had a negative impact in both metrics.

The differences in the IoU values achieved by each model can also be observed in the histograms presented in Figure 9, where it is possible to observe that models M1 and M3 achieved IoU values that lay in mostly in the 70% – 100% interval. On the other hand, the IoU values that models M2 and M4 achieved lay in mostly in the 1% – 20% interval.



(a) M1 (b) M2

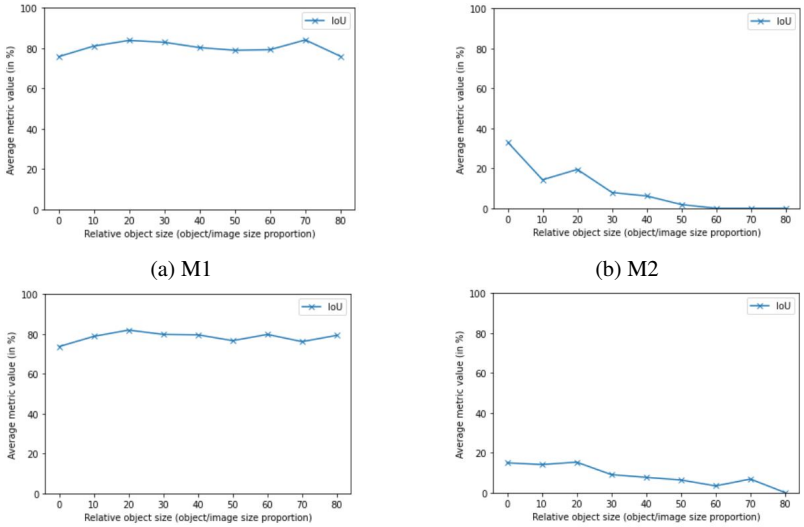
(c) M3 (d) M4

Fig. 9: Histograms of the IoU distributions achieved by each model.

6.4 Results considering variabilities in images

6.4.1 Results considering the sizes of objects

The plots presented in Figure 10 shows the performance variations associated to the relative object size in their respective images. As mentioned in Section 4, most of objects in test set are very small in relation of their respective images. This way, the performance of the models for the relatively small objects represent a large part of the overall results. Also, it is expected that in real-world detection applications, such as surveillance videos, the objects would also cover a very small portion of the images. Therefore, the results for these cases are specially important in our assessment.



(a) M1 (b) M2

(c) M3 (d) M4

Fig. 10: IoU variations associated to relative object sizes for each model.

In Figure 10, it is possible to observe that the performance of models M1 and M3 remains similar for most relative object sizes. On the other hand, model M2 and M4 present a better performance for objects having a relative size around less than 30% of the image.

6.4.2 Results considering partial occlusions

Another factor that may affect the detection performance is occlusion, which in the considered context is defined by the object being handled by a person, whose hand consequently occludes the knife blade. Table 4 compares the results between the

portion of the dataset in which the objects are partially occluded as described, and the case in which the objects are completely visible (i.e., placed in some flat surface).

Table 4: Results for partially occluded and visible knives.

Model	Occlusion	TP	FP	FN	IoU	Precision	Recall	F1-Score
M1	No	226	3	41	0.811	0.987	0.846	0.911
M1	Yes	1,830	140	58	0.773	0.929	0.969	0.949
M2	No	40	4	227	0.115	0.909	0.154	0.263
M2	Yes	1,031	770	851	0.287	0.572	0.548	0.560
M3	No	236	9	31	0.805	0.963	0.884	0.922
M3	Yes	1,828	23	60	0.750	0.986	0.968	0.977
M4	No	74	94	193	0.131	0.440	0.277	0.340
M4	Yes	653	1,117	1,235	0.142	0.369	0.346	0.357

Note that results significantly differ, especially for the M2 model, which suggests that this aspect clearly affects the models performance. Overall, all models presented better results in cases in which the object was occluded. Similar to the overall trend pointed out for the general results, models trained without transfer learning achieved better results, being M1 the best model for occluded objects and M3 the best model for non occluded objects.

6.4.3 Results considering natural illumination

Finally, another factor considered in our evaluation is the natural illumination of the scene for each image. More specifically, we compare the models results for indoor and outdoor scenes, since this change of natural illumination may present some impact in the detection performance. Table 5 summarize these results.

Table 5: Models results for both indoor and outdoor cases.

Model	Natural illumination	TP	FP	FN	IoU	Precision	Recall	F1-Score
M1	Indoor	981	139	55	0.732	0.876	0.947	0.910
M1	Outdoor	1,002	4	40	0.828	0.996	0.962	0.979
M2	Indoor	457	301	579	0.241	0.603	0.441	0.509
M2	Outdoor	576	473	466	0.301	0.550	0.553	0.551
M3	Indoor	992	15	44	0.717	0.985	0.958	0.971
M3	Outdoor	999	17	43	0.780	0.983	0.959	0.971
M4	Indoor	339	598	697	0.125	0.362	0.327	0.344
M4	Outdoor	361	613	681	0.159	0.371	0.346	0.358

According to the results, the natural illumination seems not to be a particular challenging factor for the detection models, since the results for all models tend to be similar for both indoor and outdoors scenes. It is possible to observe that the models

achieved slightly better results with outdoor scenes. In contrast with the occlusion factor, the natural illumination variations is more equally represented in the test dataset (i.e., the number of images with indoor and outdoor scenes are relatively close).

7 Conclusion

In this work, we evaluated the performance of the YOLOv4 deep neural architecture for detecting knives in natural images. In the performed experiments, two other conditions were assessed: (a) the use of a super-resolution algorithm as pre-processing step and (b) the application of a transfer learning technique. The evaluation of results not only considers the whole test dataset, but also specific subsets, in order to evaluate if there are specific conditions that can affect the results, such as object sizes, natural illumination and partial occlusions. The results have shown that using a super-resolution pre-processing algorithm only promotes better results if it is not combined with transfer learning. Moreover, the use of the proposed transfer learning technique reduced the overall performance of our YOLOv4 models.

In future works, we aim to evaluate other pre-processing techniques to be combined with new deep object detection approaches with the goal to achieve real-time processing performance, suitable for CCTV monitoring systems. Finally, we will also explore the classification aspect of object detection algorithms (i.e., including the detection of different classes of knives by considering their specific features), and extending this framework to detect also some types of firearms like guns.

Acknowledgements We acknowledge to the CYTED Network "Ibero-American Thematic Network on ICT Applications for Smart Cities", Grant No.: 518RT0559. Ángel Sánchez acknowledges to the Spanish Ministry of Science and Innovation, under RETOS Programme, with Grant No.: RTI2018-098019-B-I00. Aura Conci and Maira Moran express their gratitude to the FAPERJ, CAPES and CNPq Brazilian Agencies.

References

1. Alexey Bochkovskiy, Chien-Yao Wang, and Hong-Yuan Mark Liao. Yolov4: Optimal speed and accuracy of object detection, 2020.
2. Himanshu Buckchash and Balasubramanian Raman. A robust object detector: Application to detection of visual knives. In *2017 IEEE International Conference on Multimedia Expo Workshops (ICMEW)*, pages 633–638, 2017.
3. Alberto Castillo, Siham Tabik, Francisco Pérez, Roberto Olmos, and Francisco Herrera. Brightness guided preprocessing for automatic cold steel weapon detection in surveillance videos with deep learning. *Neurocomputing*, 330:151–161, 2019.
4. Rajib Debnath and Mrinal Kanti Bhowmik. A comprehensive survey on computer vision based concepts, methodologies, analysis and applications for automatic gun/knife detection. *Journal of Visual Communication and Image Representation*, 79, 2021.

5. Neelam Dwivedi, Dushyant Kumar Singh, and Dharmender Singh Kushwaha. Employing data generation for visual weapon identification using convolutional neural networks. *Multimedia Systems*, 28(10):347–360, 2022.
6. Tsung-Yi Lin et al. Microsoft coco: Common objects in context. In *Computer Vision – ECCV 2014*, pages 740–755. Springer International Publishing, 2014.
7. M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The pascal visual object classes (voc) challenge. *International Journal of Computer Vision*, 88(2):303–338, 2010.
8. Andrzej Glowacz, Marcin Kmieć, and Andrzej Dziech. Visual detection of knives in security applications using active appearance models. *Multimedia Tools and Applications*, 74(12):56416–56429, 2015.
9. Michał Grega, Andrzej Matiolański, Piotr Guzik, and Mikołaj Leszczuk. Automated detection of firearms and knives in a cctv image. *Sensors*, 16(1):47, 2016.
10. Michał Grega, Andrzej Matiolański, Piotr Guzik, and Mikołaj Leszczuk. Automated detection of firearms and knives in a cctv image. *Sensors*, 16(1), 2016.
11. Rida Khatoun and Sherali Zeadally. Smart cities: concepts, architectures, research opportunities. *Communications of the ACM*, 59(8):46–57, 2016.
12. Marcin Kmiec and Andrzej Glowacz. An approach to robust visual knife detection. *Machine Graphics Vision International Journal*, 20(2):215–227, 2011.
13. Christian Ledig, Lucas Theis, Ferenc Huszar, Jose Caballero, Andrew Cunningham, Alejandro Acosta, Andrew Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang, and Wenzhe Shi. Photo-realistic single image super-resolution using a generative adversarial network, 2017.
14. Roberto Olmos, Siham Tabik, and Francisco Herrera. Automatic handgun detection alarm in videos using deep learning. *Neurocomputing*, 275:66–72, 2018.
15. Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection, 2016.
16. Kang Tong, Yiquan Wu, and Fei Zhou. Recent advances in small object detection based on deep learning: A review. *Image and Vision Computing*, 97:103910, 03 2020.
17. Zhuang-Zhuang Wang, Kai Xie, Xin-Yu Zhang, Hua-Quan Chen, Chang Wen, and Jian-Biao He. Small-object detection based on yolo and dense block via image super-resolution. *IEEE Access*, 9:56416–56429, 2021.
18. Sumeth Yuenyong, Narit Hnoohom, and Konlakorn Wongpatikaseree. Automatic detection of knives in infrared images. pages 65–68, 02 2018.
19. Zhengxia Zou, Zhenwei Shi, Yuhong Guo, and Jieping Ye. Object detection in 20 years: A survey. *arXiv preprint arXiv:1905.05055*, 2019.