Brightness guided preprocessing for automatic cold steel weapon detection in surveillance videos with deep learning

Alberto Castillo, Siham Tabik, Francisco Pérez, Roberto Olmos, and Francisco Herrera

Andalusian Research Institute in Data Science and Computational Intelligence, University of Granada, 18071 Granada, Spain.
email: albertocl@decsai.ugr.es, siham@ugr.es, fperezhernandez@ugr.es, herrera@decsai.ugr.es

January 21, 2019

Abstract

The automatic detection of cold steel weapons handled by one or multiple persons in surveillance videos can help reducing crimes. However, the detection of these metallic objects in videos faces an important problem: their surface reflectance under medium to high illumination conditions blurs their shapes in the image and hence makes their detection impossible. The objective of this work is two-fold: (i) To develop an automatic cold steel weapon detection model for video surveillance using Convolutional Neural Networks(CNN) and (ii) strengthen its robustness to light conditions by proposing a brightness guided preprocessing procedure called DaCoLT (Darkening and Contrast at Learning and Test stages). The obtained detection model provides excellent results as cold steel weapon detector and as automatic alarm system in video surveillance.

Index terms— Cold steel weapon detection, Convolutional Neural Networks, video surveillance, automatic alarm system.

1 Introduction

According to World Health Organization [1], every year more than 15,000 person die in violent crimes. Around 40% of these homicides are committed with knives and sharp cold steel weapons. In video surveillance, security agents have

[1] http://www.euro.who.int/__data/assets/pdf_file/0012/121314/E94277.pdf

to visually detect the presence of weapons in the monitored scenes and make decisions in a very short time. One of the most effective solutions to this problem is to equip surveillance cameras with an accurate automatic cold steel weapon detection alarm system.

Most previous studies addressed weapon detection on X-ray, millimetric, or RGB images using classical machine learning methods [5, 7, 26, 27, 28]. Currently, the most accurate object detection models are based on deep learning techniques, particularly CNN-based models. The first work in addressing weapon detection in videos using CNNs was of Olmos et al.[16]. This work focused on pistols and was evaluated on videos of movies from the nineties.

As far as we know, the present study is the first in developing a cold steel weapon detection model using deep learning and addressing the problem of the brightness produced by cold steel surface reflectance in surveillance videos recorded in indoor scenarios. The detection of cold steel weapons in surveillance videos in indoor scenes faces several challenges:

- Cold steel weapons can be handled in different ways and a large part of the weapon can be occluded. In addition, common cold steel weapons such as knives are small and the distance between the knife and the camera can be large which makes the detection more challenging.
- The process of designing a new dataset for successfully training the detection model is manual and time consuming.
- In general, cold steel weapon, such as knives, have reflecting surface that under different light conditions can distorts and blurs their shape in the frames.
- Automatic cold steel weapon detection alarm requires the activation of the alarm in real time and needs an accurate location of the weapon in the monitored scene.

We focus on accurately detecting the most used types of cold steel weapons in crimes: kitchen knife, machete, razor, dagger and carved knives. We built a new dataset that allows the model to successfully learn the distinctive features of cold steel weapons. Then, we developed a cold steel detection model appropriate for indoor scenarios. We studied the brightness conditions that affect the detection performance and propose a new brightness guided preprocessing procedure that overcomes the problem of high brightness conditions.

The main contributions of this work can be summarized as follows:

- Build a new labeled cold steel detection database guided by the classification process.
- Analyze the best combination of CNN-based classifiers and region selection techniques for the automatic cold steel weapons detection in surveillance videos in indoor scenarios.
- Propose a new brightness guided preprocessing procedure, called Darkening and Contrast at Learning and Test (DaCoLT), for overcoming the

detrimental brightness conditions. This procedure consists of using a darkening data-augmentation technique at learning stage, and darkening plus improving contrast(with Contrast-Limited Adaptive Histogram Equalization CLAHE algorithm [19]) at inference stage.

- Develop a real time cold steel weapon detection system for surveillance videos.

Our experimental study shows that the most accurate detection model trained on our new database is R-FCN(ResNet101), providing F1 measure of 93%. The F1 obtained by the detection model in the same scenarios (i.e., same objects and actions) under different brightness conditions worsened by up to 15%. By using DaCoLT procedure we reduced this difference from 15% to 3%. In addition, the proposed cold steel weapon detection model can be used as automatic alarm system at real-time with an average activation time rate of 0.41 seconds.

This paper is organized as follows. Section 2 gives a brief analysis of the most related research studies. Section 3 gives an overview of the most influential selective search techniques and CNN-based classification models. Section 4 describes the approach we used to build our new detection database. Section 5 selects the most suitable detector for video surveillance and analyzes its performance in different brightness conditions. Section 6 describes the proposed DaCoLT procedure and analyzes its impact on the detection performance. Section 7 proves the suitability of the detector as automatic detection system using 19 scenes and AATpI metric. Finally, the conclusions are summarized in Section 8.

2 Related works

The problem of detecting a knife handled by a person in surveillance videos is closely related to (i) small objects detection in images and (ii) general objects detection using deep learning models.

The traditional area of weapon detection in images has often used classical supervised machine learning methods that require a high level of human supervision, i.e. Features from Accelerated Segment Test (FAST) [1], Scale Invariant Feature Transform (SIFT) [5], Active Appearance Models (AAMs) [7], Harris [26]. The used data are mainly X-ray or millimetric images [27, 28] for concealed weapon and RGB for visible weapons [1, 7, 8]. The authors of [7] focus on detecting the sharp point of concealed knives in X-ray images by combining a time consuming search technique with the AAMs. This approach reaches good accuracy but only on noise free X-ray images and at a high computational cost. The authors in [8] detects a knife in a RGB-images using the sliding window search approach combined with the classical SVM classifier. All these methods provide good accuracies but suffer from several limitations, they are invasive, need expensive metal-detector systems [5] such as the systems used in the airport access, cannot detect multiple weapons [11, 26] and are slow to be used in real time detection systems [1].

The state-of-the-art object detection models are based on deep Convolutional Neural Networks and showed promising results in the two most prestigious detection challenges. The most accurate detection model in the ILSVRC 2017 (Large Scale Visual Recognition Challenge) [20] reached a mean precision of around 73% [2] on a benchmark of 527,892 images arranged into 200 object classes, with an average of 2,500 images per class. The most accurate detection model on the 80 object detection benchmark in Common Objects in Context (COCO) challenge [12] also reached a mean precision of around 73%. The highest performance in COCO, a precision of 60% and recall of 80% were obtained on large objects and the lower performance, a precision of 30% and recall of 50% were obtained on small objects [3].

As far as we know, the first automatic handgun detection system based on Deep Learning was [16]. This work showed good accuracies in movies (downloaded from YouTube) with better quality, i.e., better resolution, contrast and brightness than common surveillance videos. The best results reported in this work was obtained by Faster R-CNN [18](VGGNet [21]) detection model with a rate of five frames per second(fps), which is a bit farther from being a near real time system.

3 Deep learning based detection models

The state-of-the-art detection models reformulate the detection task into a classification task following two steps. First, they apply a selective search technique to generate candidate regions from the input image then, analyze each candidate proposal with a CNN-based classification model. The combination of these two steps is critical to the detection performance.

3.1 Selective search techniques

One of the main challenges in object detection is that a priori the object of interest could be in any region in the input image. The selective search techniques intend to find the regions where the object is more likely to be located. One of the classical search techniques is the sliding window, it generates millions candidate windows, which makes this approach too slow (all these candidate windows should be feeded to the classifier) and hence not suitable for real time detection. One of the fastest search techniques is the Region Proposal (RP) search algorithm, it produces much less candidate-regions, in the order of thousands, based on the next simple assumption: the areas from the input image that contain a blobby shape are considered as potential proposals.

The first detection model in introducing the RP search algorithm was R-CNN [6]. Later, the RP search technique was converted into a fully convolutional network, called Region Proposal Network (RPN) [15], which allowed converting the two step detection process into one single step process called

[2] http://image-net.org/challenges/LSVRC/2017/
[3] http://cocodataset.org/#detections-leaderboard

end-to-end detection model. The first end-to-end model in adopting RPN as search approach was Faster R-CNN [18] followed by R-FCN [3].

Faster R-CNN operates in two steps, it uses RPN to generates around 100 regions of interest using several windows considering multiple default aspect ratios on the last feature map, then classifies these proposals with the next fully connected layers. R-FCN can be considered as an improved implementation of Faster R-CNN, it completely fuses the RPN and classifier with the aim of increasing the re-utilization of the calculation and memory accesses to shared data.

The fastest technique for object detection was included first in You Look Only Once(YOLO) [17] then in Single Shot MultiBox Detector(SSD) [13]. This technique divides the input image into a regular grid then selects candidate-regions centred in the grid-cells. The classification score of each candidate region is calculated using the scores obtained in the grid-cells to which it belongs. This technique is fast because it re-utilizes the classification of the grid cells.

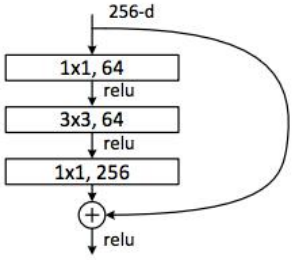
3.2 Convolutional network-based models

Convolutional Neural Networks are a particular type of neural networks, they are built as a stack of convolutional layers, pooling layers, and fully-connected layers. Convolutional and fully connected layers are learnable layers, while the pooling layer is a reduction layer that helps increasing the abstraction level between learnable layers. Recent CNNs-based classification models are increasingly showing significant improvements in a variety of computer vision tasks [14], in object recognition [22, 4], object detection [30, 16] and image segmentation [29].

In this subsection, we introduce the most influential CNN-based classification models used in our study: ResNet, Inception and Inception-ResNet-V2.

Figure 1: T

his building block is stacked several times to create ResNest50 and ResNet101, with 50 and 101 learnable layers respectively. Figure from [9].

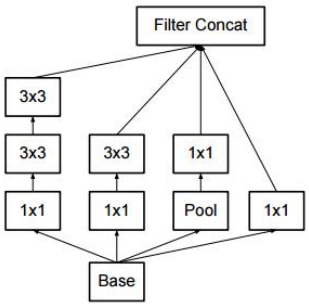


ResNet

Deep residual Network [9] (ResNet), proposed by Microsoft, won the localization and detection tasks in the ILSVRC-2015 challenge and the segmentation and

detection tasks in COCO-2015 challenge [12]. In this work, we consider two residual network architectures, ResNet50 and ResNet101 composed of 50 and 101 learnable layers respectively. Both, ResNet50 and ResNet101, are based on the building block shown in Figure 1. Very deep convolutional networks are more accurate but harder to train. ResNet is based on repeating a building block made of three convolutions, a 1x1 followed by a 3x3 and a 1x1, and a connection joining the input of the first convolution to the output of the last convolution. The connection resolve the training by fusing filtered features with original features (residual learning), as illustrated in Figure 1.

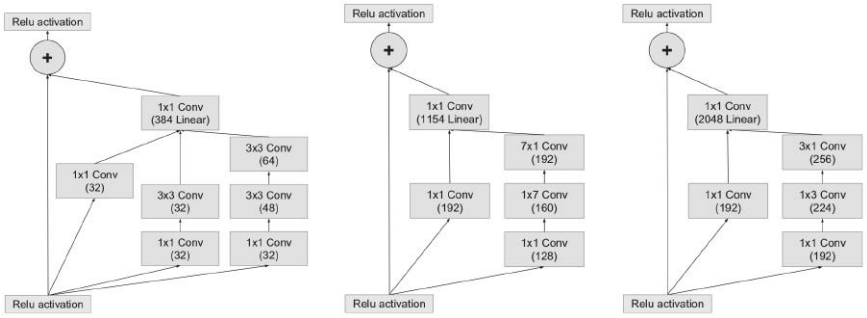
Figure 2: InceptionV2 building block. Figure from [24].



InceptionV2

GoogLeNet [24] won the ILSVRC-2014 detection challenge. It is based on the repetition of the building block module called Inception depicted in Figure 2. This module extracts different levels of features concatenated as output of module. Inception module embeds seven convolutions and one pooling layers divided into four feature channels introduced to increase width and depth. InceptionV2 network contains eight Inception modules and two refactorized modules, which are a variation of Inception.

Figure 3: Residual Inception modules used for Inception-ResNet-V2 architecture. Inception-A, Inception-B, and Inception-C blocks depicted from left to right. Figure from [23].



Inception-ResNet-V2

The combination of the residual connections and Inception, i.e. Inception-ResNet scheme, have shown better performance in image recognition. In this work, we considered the Inception-ResNet-V2 [23] architecture which currently represents the state-of-the-art in image classification. This architecture use simplified Inception modules with residual connections, as shown Figure 3. Inception-ResNet-V2 is based on variations of the original Inception module, five Inception-A, ten Inception-B, and five Inception-C block ordered from the higher to the lower network levels.

Training CNN-based classifiers

The supervised learning process of Deep CNNs, such as ResNet, Inception V2 and Inception-ResNet V2, is based on minimizing the average loss:

$$J(w) = \frac{1}{N}\sum_{i=1}^{N} L(f(w; x_i), y_i) + \lambda \mathrm{R}(w) \tag{1}$$

where x_i and y_i are the input images and the corresponding label respectively. N is the number of training examples in every iteration, L is the loss function, f is the prediction of the network using the current weights w, and R is the weight decay with the Lagrange multiplier λ. We use the Stochastic Gradient Descent (SGD) algorithm with back propagation to update the weights. SGD computes the output of the network for a set of samples, then, computes the output error and its derivatives with respect to the weights to finally update the weights of the learnable layers as follows.

$$w_{t+1} = \mu w_t - \alpha \Delta J(w_t) \tag{2}$$

where μ is the momentum weight for the current weights w_t and α is the learning rate.

The network weights can be randomly initialized if the network is trained from scratch or set to a pre-trained network weights if fine-tuning the CNN-based model. In this work we have initialized each network with the weights of the same architecture pre-trained on COCO database and retrained the last learnable layer. The selection of an appropriate combination of data-augmentation techniques should be considered to further improve the learning of the network [25].

4 Procedure for building the cold steel detection database

To build a database that allows the detection model to accurately distinguish between knives and all the objects that could be confused with knives, we first start with an initial classification dataset, Database-1, and extend it progressively with new object classes so that the number of True Positives (#TP), False

Positives (#FP), True Negatives (#TN), and False Negatives (#FN) produced by a simple classification model (VGG-16) are improved. This analysis allows us to understand which objects are critical to the learning process and consider them as background when constructing the final detection database.

We developed the database in three steps as follows:

- Database-1 includes 2 classes, the knife class contains images of knives of diverse sizes and with diverse backgrounds.
- Database-2 contains 28 classes and includes new object classes that are often present as background in the knife class in Database-1.
- Database-3 includes object classes that can be handled similarly to the knife, e.g., pen, smartphone, see four examples in Figure 4.

Figure 4: Example images of four object classes from Database-3, (a) knife class, (b) pen class, (c) mobile phone class and (d) cigarette class.

(a) (b) (c) (d)

The images used to build Database-1, -2 and -3 were downloaded from diverse websites. The characteristics of the three auxiliary databases, Database-1, -2 and -3, are shown in Table 1.

To evaluate the performance of the classification and detection models on the proposed databases, we have built two test-sets, Test-clas and Test-det.

- Test-clas is used for evaluating the classification model, it consists of 512 images, 260 images contain the knife class and 252 images contain other object classes.
- Test-det is used for evaluating the detection models, it contains 388 images, 378 contain at least one knife. Test-det includes frames taken by a surveillance IP camera (Hikvision DS-2CD2420F-IW 1080p for video, frame-rate of 30fps, field of view 95^o and MJPEG compression).

We used Keras API 2.0.4 [2] for the experiments. The performance, precision, recall, and F1, obtained by the classification model when trained on Database-1, -2 and -3 is shown in Table 2. Where,

$$Precision = \frac{\#TP}{\#TP + \#FP} \qquad Recall = \frac{\#TP}{\#TP + \#FN}$$

$$F1score = \frac{Precision \times Recall}{Precision + Recall}$$

Table 1: Databases information.

Database-	classes	total img	img knives	other img	task
1	2	1,654	598	1,056	classification
2	28	5,538	598	4,940	classification
3	100	10,039	618	9,421	classification
4	1	1,250	1,250	-	detection
Test-clas	-	512	260	252	classification
Test-det	-	388	378	10	detection

Table 2: Results of the classification model in the knife class.

Database-	#TP	#FN	#TN	#FP	Precision(%)	Recall(%)	F1 score(%)
1	181	79	174	78	69.88	69.62	69.75
2	209	51	228	24	89.70	80.38	84.78
3	213	47	228	24	**89.87**	**81.92**	**85.71**

The knife class performance has increased when extending the dataset with more object classes. The best performance is obtained when the model is trained on Database-3, but it cannot be directly used for training the detection model as the detector requires a different annotating strategy.

As final step, we built the training set, Database-4, by taking into account all the object classes, from Database-1,-2 and -3, that improve the learning because either they are handled in the same way as a knife or have similar features as a knife. Unlike in image classification, the annotation process for the detection requires indicating the object class using a bounding box. We consider two classes, the knife as the true class and the rest of objects as background. We included images of i) cold steel weapon of diverse types, shapes, colors, sizes and made of different materials ii) knives located near and far from the camera, iii) knives occluded partially by the hand, iv) objects that can be handled in the same way as knives and v) images captured in indoor and outdoor scenarios. We obtained a total number of 1,250 images. Examples from Database-4 are shown is Figure 5.

The images used to build this database were downloaded from Internet, some frames were extracted from Youtube videos and surveillance videos. In the rest of the paper we will use Database-4 for training the detection model.

Figure 5: Example images from Database-4. These images show a richer context.

5 Analysis of the deep learning approach for cold steel weapon detection

A cold steel weapon detection system in video surveillance requires a fast and robust detection model for real scenarios and under various brightness conditions. In this section, we provide:

- A description of the hardware, software setup and model hyper-parameters used to carry out the experimental study(Section 5.1).
- The performance analysis of several modern detection models to select a fast and accurate model suitable for real-time detection(Section 5.2)
- How different brightness conditions present in real scenarios affect the detection performance(Section 5.3)

5.1 Experimental setup

The detection models were built and evaluated using Tensorflow Object Detection API [10]. The experiments were carried out on a Intel Xeon E5-2630v4 CPU accelerated with NVIDIA Titan Xp GPU. A summary of the used hyperparameters is provided in Table 3. The set of data-augmentation techniques applied in the training stage are shown in Table 4.

Table 3: The hyperparameters used for tuning all detection models.

Hyperparameter	Value	Description
grid anchor generator		
scales	[0.25, 0.5, 1.0, 2.0]	List of scales for the anchors
aspect ratios	[0.5, 1.0, 2.0]	List of aspect ratios for the anchors
height-width stride	16, 16	Anchor stride in height-width dimension in pixels
height-width	256, 256	Anchor height-width in pixels
batch non max suppression		
score threshold	0.0	Scalar threshold for removing low scoring boxes
IOU threshold	0.6	Scalar threshold for removing boxes that have high IOU overlap with previously selected boxes
max detection per class	100	Maximum number of detections to retain per class
max total detections	100	Maximum number of detections to retain across all classes
score converter	softmax	Specify how to convert the detection scores
batch size	1	Required for RPN non parallel training
learning rate	0.0003	Optimizer
momentum	0.9	Optimizer
num steps	18 epochs	Number of steps to train the model

Table 4: The set of data-augmentation techniques used for training all the analyzed models.

Data augmentation technique	Parameter	Value	Description
random horizontal flip	probability	0.5	Random horizontal flip
random image scale	min ratio	0.9	Scale image by factor
	max ratio	1.4	
random RGB to gray	probability	0.1	Randomly convert entire image to gray scale
random crop image	overlap threshold	0.6	Minimum overlap threshold of cropped boxes

5.2 Selection of the best detection model

We analyzed the performance of several combinations of the state-of-the-art classification models and region selection techniques with the objective of finding the best detection model for video surveillance. In particular, we analyzed these combinations:

- SSD(InceptionV2)
- R-FCN(ResNet101)
- Faster R-CNN(Inception-ResNet-V2, ResNet50, ResNet101, and InceptionV2)

All the detection models were initialized using the pre-trained weights on COCO dataset made of more than 200,000 labeled images. We used fine-tuning by training the last fully connected layer of the network. The training process takes from three to four hours.

Table 5: Comparative analysis of the state-of-the-art detection models.

Detector	Feature extractor	#TP	#FP	Precision(%)	Recall(%)	F1(%)	fps rate
Faster R-CNN	Inception-ResNet-V2	345	0	100	91.27	**95.44**	1.3
Faster R-CNN	ResNet101	332	8	97.65	89.73	93.52	4.8
Faster R-CNN	InceptionV2	329	3	99.1	87.04	92.64	12.8
Faster R-CNN	ResNet50	326	2	99.39	86.24	92.35	4.4
R-FCN	ResNet101	335	0	100	88.62	93.97	10
SSD	InceptionV2	245	0	100	64.81	78.65	20.4

The performance of the detection models is measured in terms of true positives, false positives, precision, recall, F1, and inference time rate(frames per second). The training and testing were carried out on Database-4 and test-det respectively. In general, Faster R-CNN(Inception-ResNet-V2, ResNet101, InceptionV2, ResNet50), R-FCN(ResNet101) and SSD(InceptionV2) achieve high performance as it can be seen in Table 5. This high performance can be explained by the fact that transfer learning from COCO has been very beneficial for the learning process as COCO includes the knife class made up of around 8.500 images.

In particular, the most accurate model is Faster R-CNN(Inception-ResNet-V2) providing an F1 score of 95%. However, it is not suitable for near-real time tasks as it has a rate of 1.3 fps. The fastest detector is SSD(InceptionV2), however it produces the worst results. As we focus on video surveillance, the detection should be accurate and fast at the same time. Therefore, we select R-FCN(ResNet101) to build our cold steel weapons detector. Using 100 region proposals R-FCN(ResNet101) achieves a good precision, 100%, recall 88.62% and F1 93.97%, which is close to the best model and provides reasonable inference time rates.

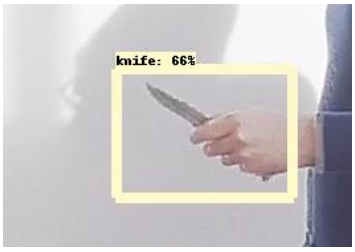
In the rest of the paper, we will interchangeably use the words inference and test and the words training and learning.

The whole detection process using R-FCN(ResNet101) in a Full HD resolution frame, of $1,920 \times 1,080$ pixels, takes $10 fps$, which is twice faster than the pistol detector proposed in [16]. This allows the cold steel weapon detection to be performed in real-time in surveillance videos.

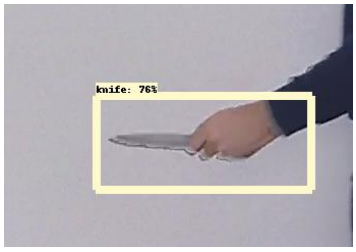
5.3 Analysis of the brightness impact on the detection performance

Under high brightness conditions, the camera sensors try to compensate these situations, but add noise or eliminate useful fine-grain details in the image, which affect the detection performance. In general, our target objects are small metallic knives that can easily disappear due to the reflection of the light in their surfaces. Next, we analyze the impact of the brightness conditions on the R-FCN(ResNet101) detection model performance.

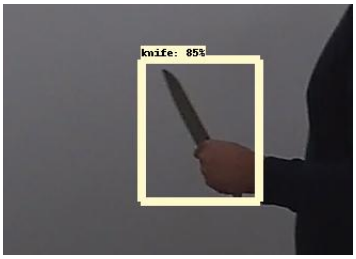
Figure 6: Results of the detection in four different brightness conditions.



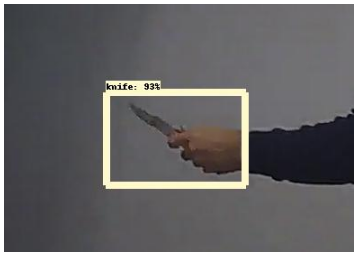
High brightness



Medium brightness



Low brightness



Artificial brightness

We used twelve test videos recorded with an IP security camera, Samsung SNH-V6410PN of resolution 1,080p, frame-rate 30fps and field of view 96.1^o. The test videos are divided into four groups of different brightness conditions, high, medium, low and artificial brightness. For a fair comparison, all the videos show the same person repeating the same actions at the same distance from the camera. All the videos were recorded using the same camera setup in the same indoor scene. The test videos include three common knives with different sizes, small, medium, and large. Small, medium and large refer to the proportion of the non-occluded part of the knife with respect to the occluded part by the hand. See examples in Figure 6. The test videos can be found through this github repository [4].

We consider a knife as ground truth when it is recognizable by the human eye. The results in terms of the total number of Ground Truth Positives #GT_P, #TP, #FP, precision, recall, and F1 in each test video are shown in Table 6. We consider a detection as TP if the overlapping between the area of the handled knife in the frame and the predicted bounding box is larger than 70%.

Table 6: Detection performance obtained on videos recorded in different brightness conditions.

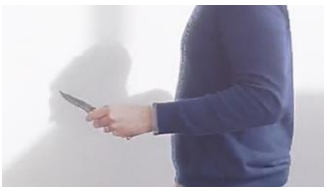
Brightness	Knife size	#frames	#GT_P	#TP	#FP	Precision	Recall	F1
High	Large	121	112	78	0	100%	69.64%	82.1%
	Medium	107	90	44	0	100%	48.89%	65.67%
	Small	137	103	53	0	100%	51.46%	67.95%
				Average		100%	56.66%	71.91%
Medium	Large	109	98	85	0	100%	86.73%	92.89%
	Medium	116	98	73	0	100%	74.49%	85.38%
	Small	138	110	64	0	100%	58.18%	73.56%
				Average		100%	73.13%	83.94%
Low	Large	126	114	104	1	99.05%	92.04%	95.41%
	Medium	114	100	70	0	100%	70%	82.35%
	Small	138	101	74	0	100%	73.27%	84.57%
				Average		99.68%	78.44%	87.44%
Artificial	Large	119	110	95	0	100%	86.36%	92.68%
	Medium	113	99	75	3	96.15%	78.13%	86.21%
	Small	96	90	65	4	94.2%	75.58%	83.87%
				Average		96.78%	**80.02%**	**87.59%**

As it can be observed from the Table 6, the performance of the detection model is unstable in a changing brightness scenario. The worst performance is obtained in high brightness conditions, and the best performance with artificial brightness. From the lower to the higher brightness conditions, the average recall decreased from 80.02% to 56.66%, and the average F1 from 87.59% to 71.91%.

Figure 7 shows an example of the detection results of very similar scenes, i.e., same pose and context, different brightness levels and contrast, but different detection results.

[4] https://github.com/alcasla/Automatic-Cold-Steel-Detection-Alarm

Figure 7: An example of the detection results in two similar situations with different brightness conditions.



High brightness

Low brightness

6 Brightness guided preprocessing: DaCoLT procedure

As shown in the previous section, the performance of the detection in surveillance scenarios is highly affected by the brightness variability. The quality of the detection model depends on the quality of the video frames. To reduce the effect of the reflectance of cold steel weapons in the frames, we developed a preprocessing method, called DaCoLT, that (1) improves the visual features (shape, texture, contrast) of cold steel weapons and (2) makes the detection model more robust to light conditions variability. DaCoLT procedure consists of two stages:

- Training the detection model on a selected range of brightness conditions using data-augmentation
- Achieving the ideal brightness condition by adjusting the darkening of the frames and improving their visual quality using a preprocessing approach before analyzing them with the detection model.

This section gives a complete analysis and assessment of DaCoLT. First, a description of DaCoLT procedure is provided in Section 6.1. The analysis of Darkening and Contrast at Test time(DaCoT) approach, which is concerned with improving the robustness of the model during test time, is given in Section 6.2 and finally the analysis of the complete DaCoLT (Darkening and Contrast at Learning and Test time) procedure is given in Section 6.3.

6.1 DaCoLT procedure

We propose a brightness guided preprocessing approach called DaCoLT to improve the robustness of the model to brightness variability at both, learning and test stages. The DaCoLT procedure can be divided into two stages, Learning and Test as described below:

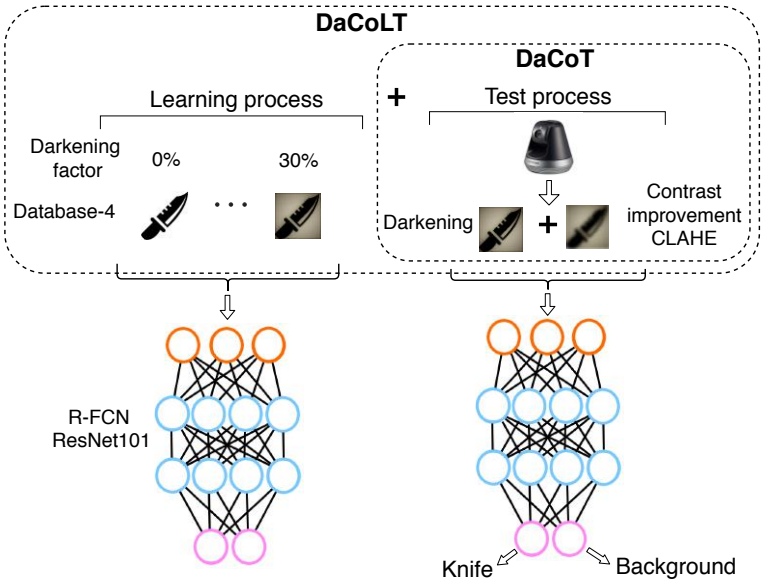
- During the learning stage, the model is trained on a specific range of brightness/darkness conditions using a darkening based data-augmentation technique. This process is as follows:

1. We first analyze and determine the effect of different brightness degrees present in the monitored environment.
2. We select the frames with the worst brightness conditions and process them by applying different darkening factors then improve their contrast. This analysis will help us to find the darkening factor that produces the best performance.
3. Finally, during the training, we apply a darkening data-augmentation technique using a darkening factor in the interval between 0% and ideal darkening factor.

- During the test stage, the highly bright frames are darkened by a specific factor, and their contrast is improved. This preprocessing stage is called DaCoT, and proceeds as follow:
  1. We first check the brightness level of each frame. If the brightness level is medium to high we darken the frame by multiplying its pixels with the corresponding darkening factor. This factor is calculated based on the difference between the ideal and the current brightness level of the frame.
  2. Afterwards, we increase the contrast of the obtained frame via CLAHE algorithm.
  3. The frame is then fed to the detection model.

The DaCoLT procedure is illustrated in Figure 8.

Figure 8: An illustration of DaCoLT procedure applied at both, learning and test time.



6.2 Analysis of darkening and contrast at test stage

To solve the instability of the detection model in variable brightness conditions, we first analyze the Darkening and Contrast at test time (DaCoT), which simulates the brightness condition that produces the best performance, i.e., low brightness and high contrast.

Table 7: Results on video frames recorded originally in high brightness conditions (i.e., worst case) when applying DaCoT.

Darkening factor	Knife size	#frames	#GT_P	#TP	#FP	Precision	Recall	F1
original high brightness	Large	121	112	78	0	100%	69.64%	82.11%
	Medium	107	90	44	0	100%	48.89%	65.67%
	Small	137	103	53	0	100%	51.46%	67.95%
				Average		100%	56.66%	71.91%
10%	Large	121	112	81	0	100%	72.32%	83.94%
	Medium	107	90	52	0	100%	57.78%	73.24%
	Small	137	103	57	1	98.28%	55.34%	71.25%
				Average		99.43%	61.81%	76.14%
20%	Large	121	112	83	0	100%	74.11%	85.13%
	Medium	107	90	55	0	100%	61.11%	75.86%
	Small	137	103	53	0	100%	51.46%	67.95%
				Average		100%	62.23%	76.31%
30%	Large	121	112	85	0	100%	75.89%	86.29%
	Medium	107	90	56	0	100%	62.22%	76.71%
	Small	137	103	53	0	100%	51.46%	67.95%
				Average		100%	**63.19%**	**76.98%**
40%	Large	121	112	80	0	100%	71.43%	83.33%
	Medium	107	90	52	0	100%	57.78%	62.6%
	Small	137	103	51	0	100%	49.51%	66.23%
				Average		100%	59.57%	70.72%
50%	Large	121	112	78	0	100%	69.64%	82.11%
	Medium	107	90	41	0	100%	45.56%	65.67%
	Small	137	103	50	0	100%	48.54%	65.36%
				Average		100%	54.58%	71.04%

The evaluation of the proposed approach under high brightness conditions when considering different darkening factors is provided in Table 7.

The performance of the detection model has improved when using a darkening factor of 30%. In average, with a darkening factor of 30% the recall and F1 have improved by 6.53% and 5.07% respectively in comparison with the obtained performance under the original high brightness condition.

The proposed preprocessing, darkening and CLAHE, takes around 29 ± 3 ms per frame on the CPU, which does not slow-down the overall detection process as this preprocessing task is performed in parallel with the detection task on the GPU. That is, the preprocessing thread is executed on the CPU and the detection thread is executed on the GPU.

This experiment allowed us to determine the range of brightness in which the detection model is unstable. We will use this information to define the darkening factor interval that improves the detection in high brightness conditions.

6.3 Analysis of darkening and contrast at learning and test stages

From the previous analysis, we found that DaCoT improves the performance of the detection model under high brightness conditions. In this section, we analyze using DaCoLT by applying different darkening levels not only at the test stage but also during the learning stage of the detection model. The used darkening data-augmentation technique consists of darkening individual training samples by randomly selecting a darkening factor in [0%, 30%].

Table 8: Results of applying DaCoT and DaCoLT on videos filmed originally under high brightness conditions and using different knife sizes, large, medium and small.

	Knife size	#frames	#GT_P	#TP	#FP	Precision	Recall	F1
original	Large	121	112	78	0	100%	69.64%	82.11%
high	Medium	107	90	44	0	100%	48.89%	65.67%
brightness	Small	137	103	53	0	100%	51.46%	67.95%
				Average		100%	56.66%	71.91%
guided brightness	Large	121	112	85	0	100%	75.89%	86.29%
DaCoT	Medium	107	90	56	0	100%	62.22%	76.71%
(Test time)	Small	137	103	53	0	100%	51.46%	67.95%
				Average		100%	63.19%	76.98%
guided brightness	Large	121	112	84	0	100%	75%	85.71%
DaCoLT	Medium	107	90	64	0	100%	71.11%	83.12%
(Learning+Test)	Small	137	103	74	0	100%	71.84%	83.61%
				Average		100%	**72.65%**	**84.15%**

Table 8 shows the impact of applying DaCoLT on the detection performance in the worst brightness conditions. The first part shows the results of the detection model on videos filmed originally under high brightness conditions and using different knife sizes, large, medium and small. The second part shows the effect of applying the proposed brightness guided preprocessing approach at the test stage, DaCoT. The third one shows the effect of the proposed preprocessing DaCoLT procedure in both, learning and test stages.

From this table, we can see that the darkening data-augmentation step included in DaCoLT improves the learning of the detection model under high brightness conditions. The average recall and F1 have respectively improved by 9.46% and 7.17% in comparison with the performance considering only ideal brightness conditions preprocessing at inference time. Particularly, the recall and F1 have improved in the harder cases by 8.89% and 6.41% respectively for the medium knife, and 20.38% and 15.66% respectively for the small knife.

Applying the brightness preprocessing during both, inference and learning steps on videos filmed under high brightness conditions improves the recall by 15.99% and F1 by 12.24% in comparison with the original high brightness conditions.

As final study, we show in Table 9 the results when applying DaCoLT procedure on videos filmed under different brightness conditions.

As it can be observed, DaCoLT improves the detection specially in the worst

conditions (i.e., highest brightness) in surveillance videos. In other words, DaCoLT allows achieving similar accuracies in the videos independently on their brightness level.

Table 9: The effect of applying DaCoLT procedure on videos filmed originally under different brightness conditions.

Brightness	Knife size	#Frames	#GT_P	#TP	#FP	Precision	Recall	F1	original F1
High	Large	121	112	84	0	100%	75%	85.71%	82.1%
	Medium	107	90	64	0	100%	71.11%	83.12%	65.67%
	Small	137	103	74	0	100%	71.84%	83.61%	67.95%
				Average		100%	72.65%	84.15%	71.91%
Medium	Large	109	98	84	0	100%	85.71%	92.31%	92.89%
	Medium	116	98	78	0	100%	79.59%	88.64%	85.38%
	Small	138	110	75	0	100%	68.18%	81.08%	73.56%
				Average		100%	77.83%	87.34%	83.94%
Low	Large	126	114	103	0	100%	90.35%	94.93%	95.41%
	Medium	114	100	74	0	100%	74%	85.06%	82.35%
	Small	138	101	72	0	100%	71.29%	83.24%	84.57%
				Average		100%	**78.55%**	**87.74%**	87.44%
Artificial	Large	119	110	95	0	100%	86.36%	92.68%	92.68%
	Medium	113	99	73	1	98.65%	74.49%	84.88%	86.21%
	Small	96	90	63	1	98.44%	70.79%	82.36%	83.87%
				Average		99.03%	77.21%	86.64%	87.59%

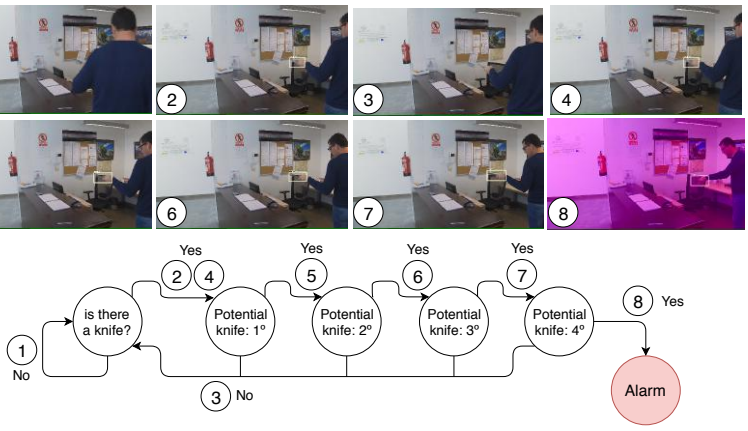
7 DaCoLT based alarm detection system

The cold steel detection model is essential for building a cold steel detection alarm system. The more robust is the detection model, the more robust is the alarm system. DaCoLT improves the visual features of knives in the frames, and hence increases the robustness and capacity of the model to detect knives correctly under variant light conditions. In this section, we show the suitability of the brightness guided procedure for cold steel knife detection alarm system using the metric AATpI (Alarm Activation time per Interval) [16].

In an automatic detection system the alarm must be activated when the system is completely confident about the presence of one or more weapons in the scene. AATpI measures the time the automatic detection alarm system takes to detect at least k successive frames of true positives. In the next analysis, we used $k = 5$ consecutive frames to activate the alarm.

Figure 9 illustrates the behavior of the alarm system using an example. The sequence of frames are analyzed frame by frame, when the system detects one isolated true positive it does not trigger any alarm. However, when the system detects five true positives in five consecutive frames, from frame 4 to frame 8, it triggers the alarm.

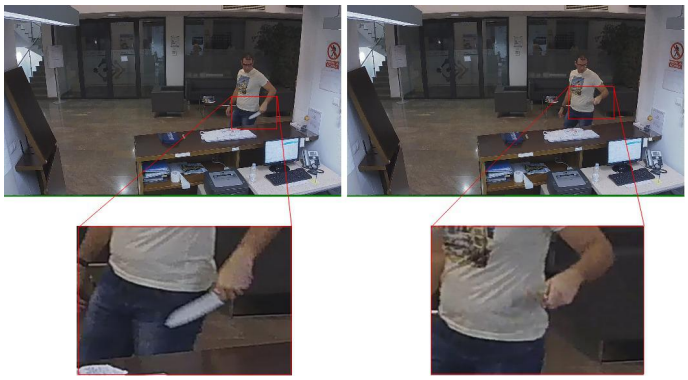
Figure 9: Alarm detection system diagram. Sample of sequence detection with alarm activation. The white box in the frames represents a true positive.



For the experimental analysis, we selected 19 scenes from diverse surveillance videos with the next requirements. Each scene is made up of at least 5 frames, recorded in a fixed scenario, i.e., in the same place, and the knife is visible for a human viewer. These scenes can be found in a public repository [5].

The model successfully detects knives in 19 scenes with an average time interval $AATpI = 0.41$s, which is good enough for an alarm system. The highest delay, 330 ms, was produced in a scene that shows noise and blur in the frames due to sudden motion of the knife, see illustration in Figure 10.

Figure 10: Examples of blurry and noisy areas (indicated by red boxes) due in part to sudden movements in two frames extracted from a surveillance video.



In summary, the proposed model has shown good performance and demonstrated to be perfectly suitable to be integrated into automatic cold steel weapon detection alarm systems.

[5] https://github.com/alcasla/Automatic-Cold-Steel-Detection-Alarm

8 Conclusions and future works

This work presents an automatic cold steel weapon detection model for video surveillance based on a new brightness guided preprocessing procedure, called DaCoLT, that further improves the quality of the detection. The obtained detection model shows a high potential even in low quality videos and provides satisfactory results as an automatic alarm system. Among 19 scenes, it successfully activates the alarm after five successive true positives in a average time of 0.41 seconds. This cold steel alarm system can be used in several applications, e.g., i) real time detection of cold steel weapon in video surveillance and ii) parental control of videos or images with violent contents.

As future work, we will address the challenging task of detecting weapons in outdoor scenarios, where moving objects can be present in the background and where adverse weather conditions can increase the difficulty of the detection.

Acknowledgments

This work was partially supported by the Spanish Ministry of Science and Technology under the project TIN2017-89517-P. Siham Tabik was supported by the Ramon y Cajal Programme (RYC-2015-18136). The Titan X Pascal used for this research was donated by the NVIDIA Corporation.

References

[1] Himanshu Buckchash and Balasubramanian Raman. A robust object detector: Application to detection of visual knives. *IEEE Multimedia and Expo Workshops*, pages 633–638, July 2017.

[2] François Chollet. Keras: Theano-based deep learning library. *Code: github. com/fchollet. Documentation: http://keras.io*, 2015.

[3] Jifeng Dai, Yi Li, Kaiming He, and Jian Sun. R-FCN: object detection via region-based fully convolutional networks. *arXiv CoRR*, abs/1605.06409, 2016.

[4] Weihong Deng, Yuke Fang, Zhenqi Xu, and Jiani Hu. Facial landmark localization by enhanced convolutional neural network. *Neurocomputing*, 273:222 – 229, 2018.

[5] Greg Flitton, Toby P Breckon, and Najla Megherbi. A comparison of 3d interest point descriptors with application to airport baggage object detection in complex ct imagery. *Pattern Recognition*, 46(9):2420–2436, 2013.

[6] Ross B. Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. *CoRR*, abs/1311.2524, 2013.

[7] Andrzej Glowacz, Marcin Kmieć, and Andrzej Dziech. Visual detection of knives in security applications using active appearance models. *Multimedia Tools and Applications*, 74(12):4253–4267, Jun 2015.

[8] Michał Grega, Andrzej Matiolański, Piotr Guzik, and Mikołaj Leszczuk. Automated detection of firearms and knives in a cctv image. *Sensors, dx.doi.org/10.3390/s16010047*, 16(1), 2016.

[9] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *CoRR*, abs/1512.03385, 2015.

[10] Jonathan Huang, Vivek Rathod, Chen Sun, Menglong Zhu, Anoop Korattikara, Alireza Fathi, Zbigniew Fischer, Ianand Wojna, Yang Song, Sergio Guadarrama, and Kevin Murphy. Tensorflow object detection api. *Code: github.com/tensorflow/models/tree/master/object_detection*, CVPR 2017 (developing).

[11] Marcin Kmieć and Andrzej Glowacz. Object detection in security applications using dominant edge directions. *Pattern Recognition Letters*, 52:72 – 79, 2015.

[12] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014.

[13] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott E. Reed, Cheng-Yang Fu, and Alexander C. Berg. SSD: single shot multibox detector. *CoRR*, abs/1512.02325, 2015.

[14] Weibo Liu, Zidong Wang, Xiaohui Liu, Nianyin Zeng, Yurong Liu, and Fuad E. Alsaadi. A survey of deep neural network architectures and their applications. *Neurocomputing*, 234:11–26, 2017.

[15] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. *CoRR*, abs/1411.4038, 2014.

[16] Roberto Olmos, Siham Tabik, and Francisco Herrera. Automatic handgun detection alarm in videos using deep learning. *Neurocomputing*, 275:66–72, 2018.

[17] Joseph Redmon, Santosh Kumar Divvala, Ross B. Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. *CoRR*, abs/1506.02640, 2015.

[18] Shaoqing Ren, Kaiming He, Ross B. Girshick, and Jian Sun. Faster R-CNN: towards real-time object detection with region proposal networks. *CoRR*, abs/1506.01497, 2015.

[19] Ali M Reza. Realization of the contrast limited adaptive histogram equalization (clahe) for real-time image enhancement. *Journal of VLSI signal processing systems for signal, image and video technology*, 38(1):35–44, 2004.

[20] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael S. Bernstein, Alexander C. Berg, and Fei-Fei Li. Imagenet large scale visual recognition challenge. *CoRR*, abs/1409.0575, 2014.

[21] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv:1409.1556*, 2014.

[22] Wenyun Sun, Haitao Zhao, and Zhong Jin. A complementary facial representation extracting method based on deep learning. *Neurocomputing*, 306:246 – 259, 2018.

[23] Christian Szegedy, Sergey Ioffe, and Vincent Vanhoucke. Inception-v4, inception-resnet and the impact of residual connections on learning. *CoRR*, abs/1602.07261, 2016.

[24] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott E. Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. *CoRR*, abs/1409.4842, 2014.

[25] Siham Tabik, Daniel Peralta, Andres Herrera-Poyatos, and Francisco Herrera. A snapshot of image pre-processing for convolutional neural networks: case study of mnist. *International Journal of Computational Intelligence Systems*, 10:555–568, 2017.

[26] Rohit Kumar Tiwari and Gyanendra K. Verma. A computer vision based framework for visual gun detection using harris interest point detector. *Procedia Computer Science*, 54:703–712, 2015.

[27] Ivan Uroukov and Robert Speller. A preliminary approach to intelligent x-ray imaging for baggage inspection at airports. *Signal Processing Research*, 4:1–11, January 2015.

[28] Zelong Xiao, Xuan Lu, Jiangjiang Yan, Li Wu, and Luyao Ren. Automatic detection of concealed pistols using passive millimeter wave imaging. In *2015 IEEE International Conference on Imaging Systems and Techniques (IST)*, pages 1–4. IEEE, 2015.

[29] Nianyin Zeng, Zidong Wang, Hong Zhang, Weibo Liu, and Fuad E. Alsaadi. Deep belief networks for quantitative analysis of a gold immunochromatographic strip. *Cognitive Computation*, 8(4):684–692, Aug 2016.

[30] Nianyin Zeng, Hong Zhang, Baoye Song, Weibo Liu, Yurong Li, and Abdullah M. Dobaie. Facial expression recognition via learning deep sparse autoencoders. *Neurocomputing*, 273:643–649, 2018.