Improved YOLOv3-tiny for Silhouette Detection Using Regularisation Techniques

Donia Ammous
National School of Engineers of Sfax, University of Sfax, Sfax
ammous.donia@gmail.com

Achraf Chabbouh
Anavid France, Road Penthièvre 10, France
chabbouhachraf.ac@gmail.com

Awatef Edhib
Sogimel: a Consulting Company in Computer Engineering and Video Surveillance, Sfax Technopole, Tunisia
awatef.edhib@anavid.co

Ahmed Chaari
Anavid France, Road Penthièvre 10, France
ahmed.chaari@anavid.co

Fahmi Kammoun
National School of Engineers of Sfax, University of Sfax, Sfax
fahmi_kammoun@yahoo.fr

Nouri Masmoudi
National School of Engineers of Sfax, University of Sfax, Sfax
masmoudi@enis.rnu.tn

Abstract: *Although recent advances in Deep Learning (DL) algorithms have been developed in many Computer Vision (CV) tasks with a high accuracy level, detecting humans in video streams is still a challenging problem. Several studies have, therefore, focused on the regularisation techniques to prevent the overfitting problem which is one of the most fundamental issues in the Machine Learning (ML) area. Likewise, this paper thoroughly examines these techniques, suggesting an improved you Only Look Once (YOLO) v3-tiny based on a modified neural network and an adjusted hyperparameters file configuration. The obtained experimental results, which are validated on two experimental tests, show that the proposed method is more effective than the YOLOv3-tiny predecessor model. The first test which includes only the data augmentation techniques indicates that the proposed approach reaches higher accuracy rates than the original YOLOv3-tiny model. Indeed, Visual Object Classes (VOC) test dataset accuracy rate increases by 32.54 % compared to the initial model. The second test which combines the three tasks reveals that the adopted combined method wins a gain over the existing model. For instance, the labelled crowd_human test dataset accuracy percentage rises by 22.7 % compared to the data augmentation model.*

Keywords: *Silhouette/person detection, GPU, loss function, convolutional neural network, YOLOv3-tiny.*

Received April 15, 2021; accepted December 14, 2022
https://doi.org/10.34028/iajit/20/2/14

1. Introduction

Deep Learning (DL), a recent technique used in video processing as well as image analysis, is characterized by a significant potential and promising result. Actually, it has been successfully used in various domains such as agriculture [1, 25, 31] (vegetation detection and identification), intelligent robotics [2, 3], video surveillance (image recognition) [7, 30], intelligent driving [16, 24] (autonomous car) and medical diagnosis [21].

Currently, person detection is an important field of CV and the first step in recognition systems. Indeed, it determines the locations of human in a source image or video sequence. Silhouette detection has played an important task in a large range of applications such as security surveillance and access control. However, it is still too costly to be implemented due to the massive computations of detection algorithms. A variety of approaches [8, 4, 24] have been developed to detect objects in images. For example, person detection is a particular case of object detection. It allows us to determine the number of people and their positions in an image if they exist. In fact, it is applied in airport security, shopping centers surveillance, apartment buildings, offices control etc., it thus represents the first step in the re-identification system. Yet, it faces a complex problem because of the great variability in people's appearance and movement. In order to obtain the best performance and resolve the existing problem, numerous studies have focused on object detection, especially in persons. A variety of Convolutional Neural Network (CNN) classifiers are used for object detection. These classifiers usually include Regional-Convolutional Neural Network (R-CNN) [9, 10, 30], DPM (deformable part model) [11, 26], Single Shot MultiBox Detector (SSD) [19] and YOLO [12, 49].

The rest of this paper is organised as follows. YOLOv3-tiny structure is described in section 2. The regularisation techniques in Machine Learning (ML) are investigated in section 3 and the proposed approach is introduced in section 4. The experimental results are discussed in section 5. Finally, conclusions are drawn in section 6.

2. Related Works

R-CNN [9] is accurate but not fast enough on the PASCAL VOC 2007 dataset. Fast Region-based Convolutional Network (Fast R-CNN), the next version of R-CNN, was developed by Girshick *et al*. [10] in order to reduce time consumption for the high number of models required to analyse all the regional proposals. The latest object detection method, the Faster R-CNN [30], was then developed to further decrease the detection network uptime. Despite its high accuracy rate, this classifier characterized by a heavy computational cost cannot be applied in many real time applications since they are carried out in computationally-limited platforms. The YOLO model has surmounted this problem through improvements in both speed and accuracy. This model, with its various versions, has been proven to be highly efficient in object detection, especially in silhouette detection. Base YOLO [12], also called YOLO Version 1 (YOLOv1), processes images at 45 Frames Per Second (FPS) which is twice to nine times faster than that of Faster R-CNN. YOLO Version 2 (YOLOv2) [33] is a variously improved model of YOLO which maintains the advantage of speed and tries to rise the Mean Average Accuracy (MAP) value from YOLOv1. YOLOv3-tiny [50] is a simplified model of YOLOV3 [34] as it reduces the depth of the convolutional layer. It is used as it is faster than YOLOv3. Furthermore, YOLOv3-tiny is seen as the best real-time object detection system compared to other algorithms like SSD and YOLOv3. Despite their high detection accuracy, these latters ability to perform real-time object detection on low performance devices or PCs is far from being satisfactory. In addition, YOLOv3-tiny learns general representations of objects, outstripping other detection methods including DPM [11, 26] and R-CNN. YOLOv4-tiny is one type of object detection methods of deep learning. It is designed based on Yolov4 method. This method used feature pyramid network to extract feature maps without requiring the spatial pyramid pooling and path aggregation network that are adopted in Yolov4. The Yolov4-tiny simultaneously uses two different scale feature maps: 13x13 and 26x26 to estimate the detection results [12, 18]. In [14], a study comparative were performed in YOLOv3-tiny, YOLOv4-tiny to test their performance in FPS. FPS is a measurement of the number of images that can be accurately detected in one second. The speed of object detection for Yolov3-tiny can reach 277 Frames per second. Yolov4 tiny have complex network structure and many parameters than yolov3-tiny. As a result, it has worse performance in fps (see Table 1).

Table 1. Comparison of different methods in FPS [14].

Method	FPS
Yolov3-tiny	277
Yolov4-tiny	270

Prasetyo *et al*. [27] used Binary Floating-Point Operations (BFLOPS) to find out the model computation volume. In [27], the data presented in the following Table shows that the yolov4-tiny model uses more computation volume because additional layer in your architecture (see Table 2). As a result, yolov3-tiny has faster object detection than yolov4-tiny. It is more suitable for real-time object detection, especially for developing on embedded devices.

Table 2. Model comparison [27].

Model	BFLOPS
Yolov3-tiny	5.45
Yolov4-tiny	6.79

Since the FPS of YOLOv4-tiny performance is lower than yolov3-tiny and their BFLOPS performance is elevated. We choose YOLOv3-tiny for our application which requires higher image processing speed.

Yang *et al*. [49] apply the well-known deep learning detection algorithm to accomplish tiny-face identification and incorporate the Yolov3-tiny model with Dropblock method. Dropblock is a regularization technique used in the convolutional layer, where the activation units are spatially connected. The feature map's contiguous region's neurons would be dropped by dropblock, forcing the neural network to pick up new features. Xun *et al*. [48] suggested an enhanced Mish-L2-multitask Convolutional Neural Network (MTCNN) model based on the MTCNN model to further boost the accuracy of small-size face detection. The P-Net CNN's maximum pooling layer was first taken off. Second, a regularization term was included in the crossentropy loss function. Finally, Mish activation functions were used to replace the activation functions that were applied to three subneural networks of the MTCNN model. Jamiya and Rani [15] provided a real-time vehicle detection method using the YOLOv3-tiny network and a lightweight deep neural network model named LittleYOLO-SPP. The feature extraction network of the YOLOv3-tiny object detection network is changed to improve the speed and precision of vehicle detection. In order to improve the network's capacity for learning, the suggested network included a technique called spatial pyramid pooling, which comprises of several scales of pooling layers for concatenation of features. Niu *et al*. [23] introduced four offline data augmentation techniques to enhance the effectiveness of CNN on defect detection of sanitary ceramics. In order raise the original dataset's quality, Niu *et al*. [23] proposed data augmentation technique. Image generation, image mosaic, image fusion and image rotation mosaic are all used in this case. Zhang *et al*. [53] suggested convolutional layer channel pruning to train effective deep object detectors. With the goal of obtaining “slim” object detectors. The authors imposed L1 regularization on channel scaling factors to enforce channel-level sparsity of convolutional layers and prune less informative feature channels. The researchers provided SlimYOLOv3 as a potential solution for real-time object recognition on UAVs since it has less trainable parameters and Floating Point Operations (FLOPs) than the original YOLOv3. Qi *et al*. [29] suggested a brand-new detection network architecture called MYOLOv3-Tiny, while depthwise and pointwise convolution are used to lower the computational complexity of the network and the backbone network is designed using a linear bottleneck structure with inverted residual to effectively extract the fasteners' features. For pedestrian and vehicle detection in traffic monitoring, an improved YOLOv3-tiny model is suggested [39]. The YOLOv3-tiny model's backbone network structure was altered, deep detachable convolution operation was added, and the network's fundamental residual block unit was created, all of which improved the backbone network's capacity for feature extraction. Compared to the original model, the improved YOLOv3-tiny model offers a greater measurement accuracy. The detection speed is 150 FPS when this model is applied to the 1080P traffic video on the NVIDIA RTX 2080, which is completely capable of real-time detection. The suggested model [17] uses the Dual-Path Network (DPN) module and the fusion transition module to efficiently extract features. It also uses a dense connection strategy to enhance multi-scale prediction, which allows it to accurately classify and locate objects.

In general, we note that the use of deep learning techniques for person detection is still restricted to specific conditions including camera position and orientation. Despite the different algorithms developed for the detection of people and the various studies conducted on this subject, numerous constraints are faced to detect people. These roadblocks include the position and orientation of the cameras (in the front, at the top,...), as well as the detection medium characteristics. For instance, detecting people in a mall or in a store where there are clothing mannequins that have the morphology of a person leads to changeable test results that differ from one environment to another. Hence, this study attempts to produce a more generalized detection model. The main contributions of this paper can be summarized as follows:

- A several classifiers for object/silhouette detection are presented.
- YOLOv3-tiny is adopted duo to this detectors is faster than the traditional systems. (and other classifiers)
- The benefits of regularisation techniques are explained in this work.
- The proposed approach adds a dropout layer in the existing YOLOv3-tiny architecture and changes hyperparameters in order to ameliorate silhouette detection. In addition, data augmentation techniques are used to get a large dataset for training.
- A comparative study is realised in order to validate the effectiveness of the combined method for person detection.
- The method put forward has achieved a good performance relatively to tiny-yolov2, tiny-yolov3 and tiny-yolov4 original models.

3. Detailed Description of YOLOv3-tiny

Compared to traditional systems, YOLO is regarded as a new, faster approach designed for object detection. YOLO-tiny detection system is adopted because of its speed and real-time application (see Table 3). This object detection algorithm, based on Deep Learning, was developed by «Redmon *et al*. [32]» at the University of Washington in 2015. It is evaluated on the basis of data PASCAL VOC allowing the detection of twenty classes:

- Person.
- Animal: bird, cat, cow, dog, horse, sheep.
- Vehicle: plane, bicycle, boat, bus, car, motorcycle, train.
- Interior: bottle, chair, dining table, potted plant, sofa, TV/ monitor.

Table 3. Other models performances.

Test database	Network Structure	MAP (%)	Time (ms)
COCO-test	Fast R-CNN [10]	39.9	2000
	Faster RCNN [30]	42.7	142
	Faster RCNN Resnet 101 [13]	48.4	200
	SSD 300 [19]	43.1	23
	SSD 512 [19]	48.5	45
	DSSD321 [12]	46.1	105
	YOLOV2 [14]	44	15
	RefineDet320 [27]	49.2	25
	YOLOv3-tiny origin [49]	33.1	4.5

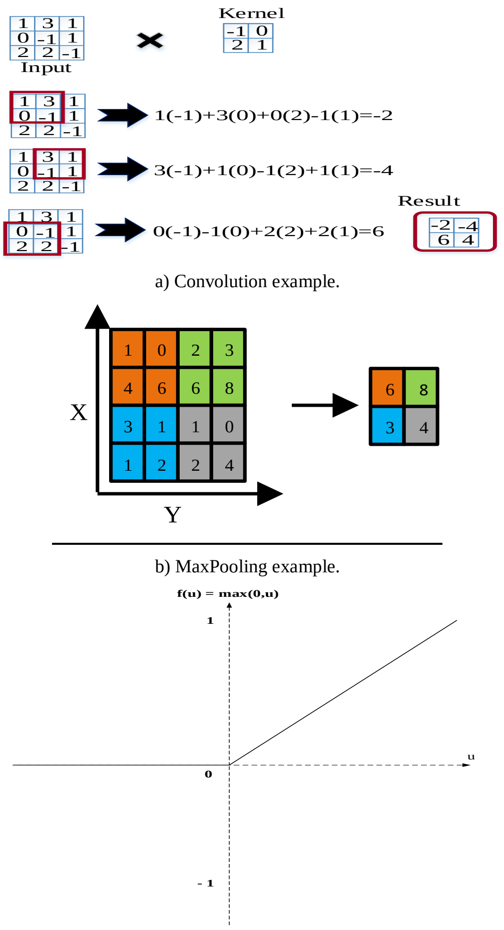
YOLO has 24 convolution layers followed by 2 Fully Connected (FC) layers (see Table 4).

Table 4. The YOLOv3-tiny architecture network.

Layer	Layer type	Filter	Size\Stride	Imput	Output
0	convolutional	16	3×3⁄1	416×416×3	416×416×16
1	maxpool	__	2×2⁄2	416×416×16	208×208×16
2	convolutional	32	3×3⁄1	208×208×16	208×208×32
3	maxpool	__	2×2⁄2	208×208×32	104×104×32
4	convolutional	64	3×3⁄1	104×104×32	104×104×64
5	maxpool	__	2×2⁄2	104×104×64	52×52×64
6	convolutional	128	3×3⁄1	52×52×64	52×52×128
7	maxpool	__	2×2⁄2	52×52×128	26×26×128
8	convolutional	256	3×3⁄1	26×26×128	26×26×256
9	maxpool	__	2×2⁄2	26×26×256	13×13×256
10	convolutional	512	3×3⁄1	13×13×256	13×13×512
11	maxpool	__	2×2⁄2	13×13×512	13×13×512
12	convolutional	1024	3×3⁄1	13×13×512	13×13×1024
13	convolutional	256	3×3⁄1	13×13×1024	13×13×256
14	convolutional	512	3×3⁄1	13×13×256	13×13×512
15	convolutional	255	3×3⁄1	13×13×512	13×13×255
16	yolo	__	__	__	__
17	route	__	__	__	__
18	convolutional	128	1×1⁄1	13×13×256	13×13×128
19	up-sampling	__	2×2⁄1	13×13×128	13×13×128
20	route	__	__	__	__
21	convolutional	256	3×3⁄1	13×13×384	13×13×256
22	convolutional	255	1×1⁄1	13×13×256	13×13×256
23	yolo	__	__	__	__

The entry is a 416*416*3 size image. The first convolution layer extracts the characteristics of the image while the last convolutional layer and the two F C layers predict the scores. This network consists essentially of convolution layers, pooling layers, correction layers (Rectified Linear Unit (*ReLU*)) and Fully-Connected layers. As the key component of the network, the convolution layer aims at identifying the presence of a set of characteristics in the input image. The convolution principle consists in dragging a window on the image and calculating, each time, the convolution product between the filter and a block of the scanned image. Figure 1-a) shows an example of convolution. The pooling layer is placed between two convolution layers. It receives as input the feature maps resulting from the convolutional layer. It reduces the size of the images, while preserving their important characteristics. The input image is split into regular cells to keep the maximum value within each cell. An example of pooling is illustrated in the Figure 1-b).

ReLu denotes the nonlinear function defined by max (0, u). This layer replaces all negative values received as inputs with zeros. It plays the role of activation function. The appearance of this function is shown in Figure 1-c). The FC layer represents the last layer of the network. It receives a vector as input and produces as outputs a new vector. Neurons in a FC layer have connections to all outputs of the previous layer.



a) Convolution example.

b) MaxPooling example.

c) The ReLU Function Shape.

Figure 1. Description of layers.

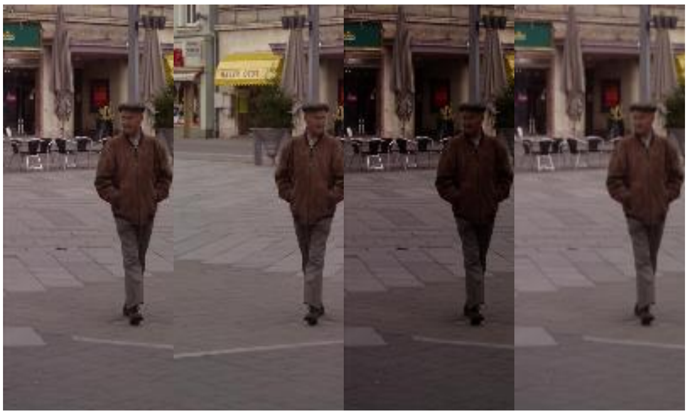
4. Regularisation Techniques in ML

Overfitting is one of the most basic problems in the ML field. Several researchers have developed different techniques to overcome this problem. In fact, the model performs well in the training dataset but poorly in the novel and unknown one. Several techniques are described below to prevent overfitting [28, 51].

- Simplifying the Model It involves reducing the network complexity by removing layers or reducing the number of neurons in hidden ones.
- Early Stopping Early Stopping is based on the calculation of the loss and MAP. MAP is calculated using a validation dataset to estimate the performance of the proposed model, to stop the training process in the best weight.
- Use Data Augmentation Data augmentation is applied to expand and enrich one’s own dataset. This advantage allows the network to learn any dataset. In this paper, three data augmentation techniques are applied. In fact, Figure 2 illustrates the various reformations of images using mirror, contrast and blur techniques of data augmentation.
- Use Dropouts the FC layers occupy most of the CNN memory. Moreover, the FC concept creates an exponential problem of memory called “overfitting”. To avoid this problem, the dropout method randomly inactivates neurons with a predefined probability in the FC layer [37]. Dropout is thus a regularization technique that modifies the network architecture.

5. Improved YOLOV3-Tiny Model

In this section, the process of YOLOV3-tiny training is explained and the proposed approach that addressed the challenging problem of silhouette detection is described.



a) Origin. b) Mirror. c) Contrast. d) Blur.

Figure 2. Data augmentation techniques.

5.1. Process of YOLOV3-tiny Training

The steps undertaken for training YOLOv3-tiny are reported to detect persons in images (see Figure 3).The first and foremost task for the training is the preparation of a database and their annotations. Indeed, a database containing 204960 images has been prepared using data augmentation methods (see Figure 2). The entire database divided into training set (90%) and test set (10%). In secod step, the necessary files for the training are prepared. These files are "obj.data" which incorporates information about detection and important paths, "obj.names" bearing the detection classes for our case class 'person' and "obj.cfg" representing the configuration file whose parameters are to be modified according to our needs (batch, subdivision, classes...). As a result the training process will be lanched. Two files generated from training as output the trained model and the log file. The log file contains the loss in each batch. Once the loss becomes below a threshold (example: 0.0001), the training is immediately stopped.

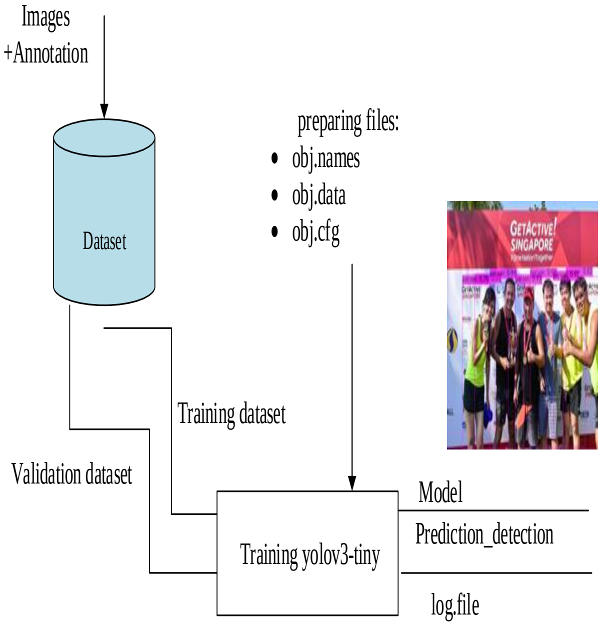


Figure 3. Training principle.

5.2. Proposed Approach

In the first part of the experimental test, the Anavid [40] dataset, which includes 97073 images, is collected. Thanks to the technique of data augmentation, a large dataset of 204960 images is obtained, preventing the overfitting phenomenon. Training is realised on a one-class "person". The MAP of the final weight generated from training in different test datasets Voc2007-test (2097 images), COCO-test2017 (2693 images), PennFudanPed (170 images) [41] is computed. In the second part of the experimental test, a dropout layer is added into the original architecture of YOLOV3-tiny (see Figure 4) to further improve the model performance. The best weight MAP generated from training in different test datasets is calculated [42].

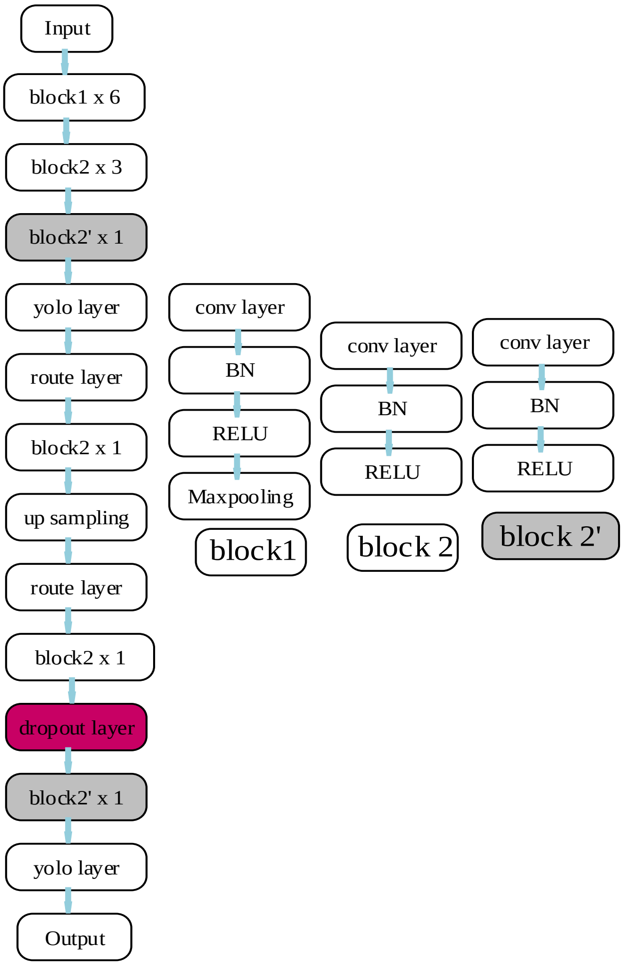


Figure 4. The Proposed YOLOv3-tiny Network Architecture

In addition, the same dataset is treated in the YOLOV3-tiny and the following hyper-parameters are adjusted (Table 5):

Table 5. Parameter of our YOLOV3-tiny network.

size of input	416 x 416
Batch	64
Subdivision	4096
Steps	2200160, 2350180
learning rate	0.001
burn_in	2000
max_batches	2500200
Classes	1
filter	18
Random	1
Anchor	7, 23, 17, 54, 27, 107, 63, 98, 39,179, 67, 240, 196, 161, 123, 327, 279, 350
ignore_thresh	0.9
iou_normalizer	0.5
iou_loss	Giou
Probability	0.5
Dropblock	1

6. Experimental Results

In this section, we evaluate two of our regularization methods which boost the network performance and avoid the overfitting of the model by some metrics of evaluation. Furthermore, we explaine several ways of our method to improve the model generalization ability.

6.1. Evaluation Metrics

- Loss function: a loss calculation, also called sum-square error [35, 52], is a simple addition of differences, including coordinate errors, IOU errors, and the classification error. The loss function can be expressed by the following formula:

$$loss = \sum_{i=0}^{S^2} \text{coordErr} + \text{iouErr} + \text{clsErr} \qquad (1)$$

Where coordErr refers to the coordinate errors, iouErr to the IOU errors and clsErr to the classification error.

- Intersection over Union (IOU): IOU is calculated by the overlap zone between the predicted bounding box and the truth bounding box divided by the union zone between them.

$$IOU = \frac{area\ of\ overlap}{area\ of\ union} \qquad (2)$$

Precision: Precision is the ability of a model to identify only relevant objects. This is the percentage of correct positive predictions and is given by the following equation:

$$precision = \frac{TP}{TP+FP} \qquad (3)$$

Where True Positive (TP): the prediction is true, the number of people who are correctly detected and FP (False Positive): The prediction that is false for which IOU<0.5.

- Recall: Recall relates to a model ability to find all relevant cases. This is the percentage of true positives detected among all relevant field truths and is given by:

$$recall = \frac{TP}{TP + FN} \qquad (4)$$

Where False Negatives (FN): the prediction that is false, the number of people who are considered negative while they are positive.

- AP: Average Precision (AP) is a measurement commonly used to calculate object detectors accuracy. It works out the mean precision value for the Recall value between 0 and 1.
- MAP: MAP is an average AP value for multiple verification sets. It is used as an indicator to measure detection accuracy in target detection.

A MAP metric frequently exists in the deep learning area. In order to calculate MAP, a series of precision recall curves with the IOU threshold set is drawn (see Figure 5). The different lines are drawn by changing thresholds starting from 0.5 to 0.95. The red line is drawn with the highest IOU (0.95) and the orange line is drawn with the smallest one (0.5). For each line, AP value is the area enclosed by the precision-recall curve and the coordinate axis. The metric calculates the Average Accuracy value (AP) for each class across all IOU thresholds so as to draw the P-R curve. Then, to obtain the MAP value of the entire model, the metric calculates the average value of the AP values for all classes.

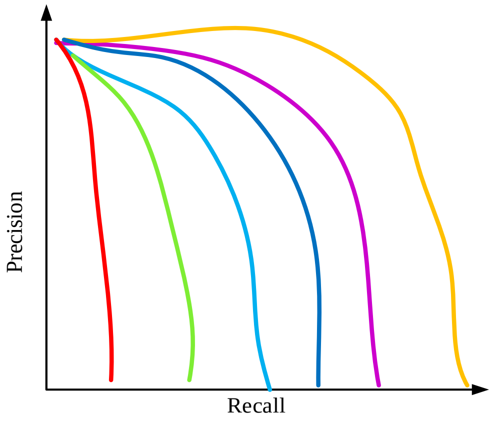


Figure 5. MAP precision-recall curves.

- Accuracy: A accuracy is calculated by the following formula:

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \qquad (5)$$

Where True Negatives (TN): the number of negative examples which are correctly rejected.

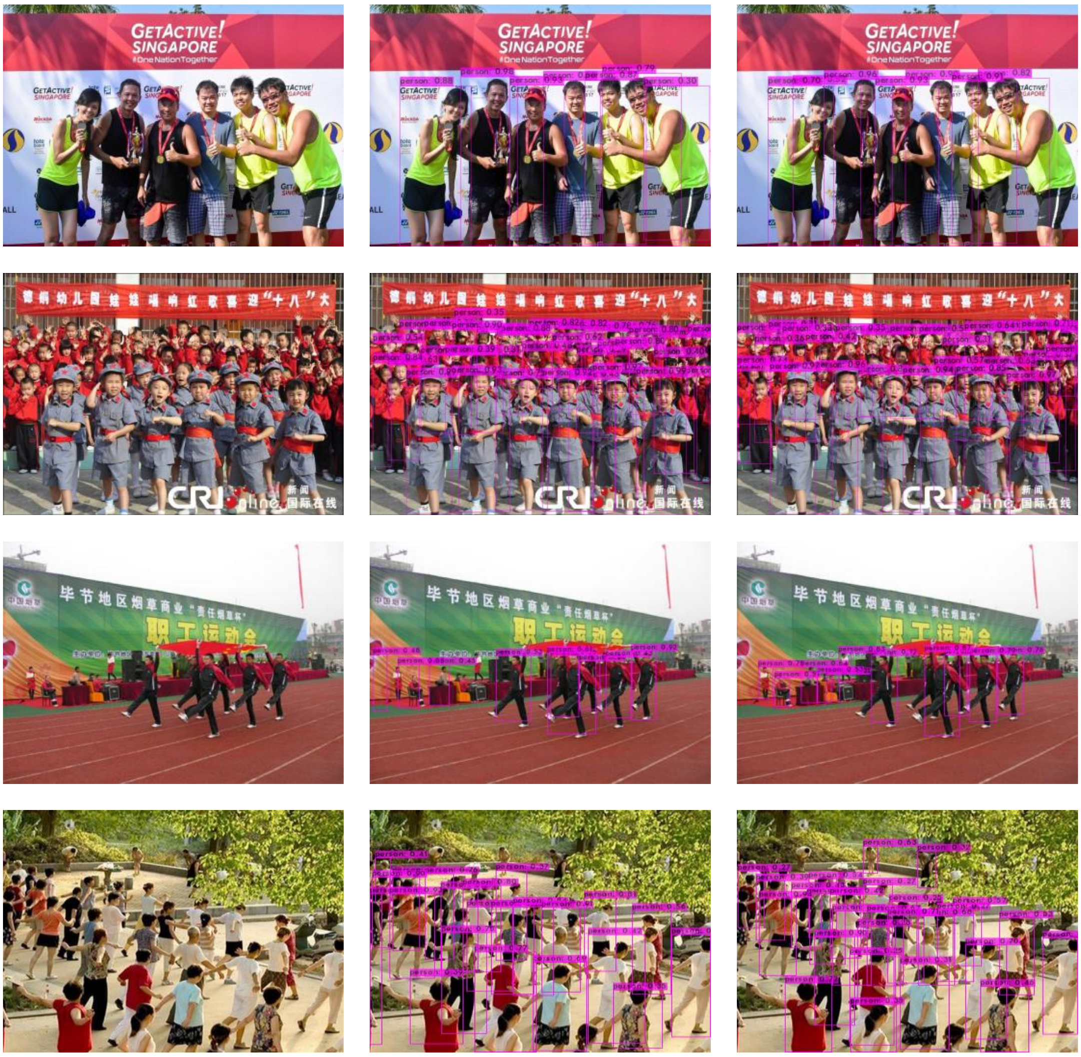
6.2. Evaluation Results

Nowadays, deep neural networks algorithms are mainly improved by using various regularisation techniques such as data augmentation, dropout and early stopping. The experiments present two regularisation techniques. Therefore, the experimental results consist of two parts. We begin by comparing the results of the data augmentation technique (see Table 6). The experiments are performed on the Ubuntu 16.04 system. Under the PyCharm development environment, the program of labels reformulation is written in Python 3.6. The YOLOv3-tiny algorithm is run under the darknet framework [36]. YOLO is an open source with different versions [50]. The machine server contains a processor which is an Intel Core i7-8750H Central Processing Unit (CPU) @ 2.20GHz×12 and two Graphics Processing Unit (GPU) tesla v100 have 32 Go memory. Experiments are conducted in one dataset collected from real video surveillance of 97073 images for training. To verify the performance of the model trained, different test dataset, mainly Pascal Visual Object Classes (PASCAL VOC) [6, 43], Microsoft Common Objects in COntext (MS COCO) [20, 44], crowdhuman [38, 45] and PennFudanPed [46, 47], is employed. In order to verify the performance of the model trained on the collected dataset with data augmentation, we compare the accuracy with the existing YOLOv3-tiny model using one’s own collected datasets (see Table 6). Compared to the origin YOLOv3-tiny model, our model has achieved a gain of 32.54 %, 16.38 % and 14.14 % using VOC, COCO and PennFudanPed test dataset respectively. Moreover, the variety of poses, appearance, textures, colors, background clutter, body shape and style of clothing possessed by these datasets further prove the importance as well as the outstanding performance of our method.

Table 6. Data augmentation gain.

	Accuracy (%)		Gain (%)
Test database	Model trained on the collected dataset	Model trained on the collected dataset with data augmentation	
COCO-test 2017 (2693 frames)	11.73	28.11	16.38
PennFudanPed (170 frames)	74.03	88.17	14.14
Voc2007-test (2097 frames)	18.4	50.94	32.54

In the second part of the experimental results, a novel model using the same dataset 204960 images is trained. Some changes to YOLOV3-tiny architecture are also made. In fact, a dropout layer is added before the last convolution layer in the network with some adjusted parameters (see Table 5). This approach helps better the quality of the detection silhouette algorithm (see Table 7). The proposed method is compared to the model trained on the collected dataset with data augmentation. The comparisons with data augmentation models demonstrate the effectiveness of the proposed method. In fact, a gain of 21.16 %, 2.05 %, 18.68 % and 22.7 % is obtained using COCO, PennFudanPed, VOC and crowd_human test dataset respectively. In fact, dropout is used to regularize the Deep Neural Networks (DNN) algorithm. Input elements are randomly set to zero in the network with a probability of 0.5. Other weight elements are rescaled. Consequently, each node is used independently and it is not necessary to rely it on the output of other nodes. Indeed, dropout reduces the complex co-adaptations of neurons since the latter cannot rely on other neurons. In traditional networks, any error in the neural networks structure propagates during training. Yet, using a novel architecture leads dropout to lower error rates. The performance of a neural network and the rate of accuracy hence increase.



Origin image. Image with data augmentation. Image with dropout and data augmentation.

Figure 6. Bounding box visualisation.

Table 7. Gain of the proposed combined method.

	Accuracy (%)		Gain (%)
Test database	**Model trained on the collected dataset with data augmentation**	**modified model (dropout+hyperparameters) trained on the collected dataset with data augmentation**	
COCO-test-2017 (2693 images)	28.11	49.27	21,16
PennFudanPed (170 images)	88.17	90.22	2.05
Voc2007-test (2097 images)	50.94	69.62	18,68
crowd_human_test (4370 images)	23.58	46.28	22.7

In this paper, it can also be seen from experiments for silhouette detection, more detailed detection results (Figure 6). For line 1 of Figure 6, confidence percentages increase in certain bounding boxes from (79% 87% 80% 93% 98% 88%) to (82% 91% 90% 93% 96% 70%). For line 2 of Figure 6, the parts of the people in the border of the image are detected by the combination of the two techniques of regularization including data augmentation and dropout. For lines 3 and 4 of Figure 6, bounding boxes are added in both cases of images of line 3 and 4 (different camera position and orientation) combining the two regularization techniques. So, our method improves silhouette detection and is robust for any dataset. Consequently, it is standard as it boosts the generalization ability of the model. The two tests (data augmentation and dropout) are complementary. Each has its effect on the learning algorithm YOLOv3-tiny.

6.3. Comparison with Previous Works

The combined method has achieved good performance. Indeed, the MAP value has reached 49.27%, which is 9.47 % and 16.09 % higher than both of tiny-YOLOv4 origin and tiny-YOLOv2 respectively, which is more than that of tiny-YOLOv3 origin for COCO test dataset (see Table 8). For VOC test dataset, the result has been ameliorated by 32.49 %, 18.06 %, 12.32 % compared to the YOLOv3-tiny origin, YOLOv2-tiny and YOLOv4-tiny origin. As a result, the proposed model with the two regularization techniques has accomplished the highest accuracy rate as compared to YOLOv2-tiny and YOLOv3-tiny origin. The rich training dataset obtained as well as the integration of the dropout layer into YOLOv3-tiny architecture enhance the detection accuracy.

Due to the diversity of the characteristics of the images in the data set, different accuracy rates are obtained in publicly available datasets, mainly COCO-test, VOC-test, crowd_human_test and Pen-test. In fact, Penn-Fudan database is dedicated for pedestrian detection. Penn-Fudan is a simple image database containing 170 images taken from scenes around campus and urban street. Hence, the background of the frames is often the same. All pedestrians are walking, thus, they are straightened up (full-body appearance) and detected clearly. Therefore, the network is more responsive and can learn more quickly. Furthermore, the accuracy value reflects a significant improvement. However, the background of the crowd_human_test dataset is complicated, the postures of the human beings are different, the degree of occlusion and the size of the human bodies are not the same. The value of accuracy is very low compared to Penn-Fudan database. This is up to dataset nature. Some persons in the dataset are partially visible persons and the position of the camera is far in some images. In crowded scenes, due to the high pedestrian density and high occlusion, most existing pedestrian detection methods are unable to obtain a good result [5], and have poor robustness to small targets. In fact, when a target person is largely overlapped with other persons, the detector may fail to identify the boundaries of each pedestrian since they have similar appearances. Thus, the detector may shift the target bounding box to other persons mistakenly or treat the crowd as a whole.

Table 8. Comparison of proposed method with other detectors.

Base de test	Network structure	Accuracy (%)	Gain (%)
COCO-test	**YOLOv3-tiny origin**	20.64	28.63
	YOLOv2-tiny	33.18	16.09
	YOLOv4-tiny-origin	39.8	9.47
	combined method	49.27	__
Pen-test	**YOLOv3-tiny origin**	80.30	9.92
	YOLOv2-tiny	84.68	5.54
	YOLOv4-tiny-origin	85.7	4.52
	combined method	90.22	__
voc-test	**YOLOv3-tiny origin**	37.13	32.49
	YOLOv2-tiny	51.56	18.06
	YOLOv4-tiny-origin	57.3	12.32
	combined method	69.62	__
crowd_human_test	**YOLOv3-tiny origin**	6.72	39.56
	YOLOv2-tiny	16.63	29.65
	YOLOv4-tiny-origin	27.6	18.68
	combined method	46.28	__

7. Conclusions

In this paper, we propose a novel method for silhouette detection so as to improve the YOLO-tiny V3 learning algorithm. The silhouette detection system is enhanced by integrating the data augmentation technique into a new YOLO-tiny v3 neural network architecture and modifying the configuration file hyperparameters.

Experimental results show that the proposed method can significantly improve person detection accuracy. Thanks to data pre-processing used, we provide a high accuracy, outperforming conventional models. Moreover, study comparisons of specific models prove that the combined technique has contributed to a great accuracy improvement. In future works, we will deepen in the study of existing methods that improve the quality of object detection like other techniques of data augmentation.

Acknowledgment

This work was performed under the MOBIDOC PROMESSE fund from the Tunisian ministry of higher education and scientific research. This research publication is supported by ANAVID France. I would like to express my sincere gratitude to ANAVID Company and LETI research laboratory. I am grateful for the constructive feedback from the reviewers.

References

[1] Ayadi S., Ben Said A., Jabbar R., Aloulou C., Chabbouh A., and Achballah A., "Dairy Cow Rumination Detection: A Deep Learning Approach," *in Proceedings of International Workshop on Distributed Computing for Emerging Smart Networks*, Bizerte, pp. 123-139, 2020.

[2] Al-Sa'd M., Al-Ali A., Mohamed A., Khattab T., and Erbad A., "RF-Based Drone Detection And Identification Using Deep Learning Approaches: An Initiative Towards A Large Open Source Drone Database," *Future Generation Computer Systems*, vol. 100, pp. 86-97, 2019.

[3] Ammous D., kallel A., Kammoun F., and Masmoudi N., "Analysis of Coding and Transfer of Arien Video Sequences from H. 264 Standard," *in Proceedings of 5th International Conference on Advanced Technologies for Signal and Image Processing*, Sousse, pp. 1-5, 2020.

[4] David B. and Rangasamy D., "Spatial-Contextual Texture and Edge Analysis Approach for Unsupervised Change Detection of Faces in Counterfeit Images," *International Journal of Computers and Applications*, vol. 37, no. 3-4, pp. 143-159, 2015.

[5] Dong X., Han Y., Li W., and Li B., "Pedestrian Detection in Metro Station Based on Improved SSD," *in Proceedings of IEEE 14th International Conference on Intelligent Systems and Knowledge Engineering,* Dalian, pp. 936-939, 2019.

[6] Everingham M., Van Gool L., Williams C., Winn J., and Zisserman A., "The Pascal Visual Object Classes (Voc) Challenge," *International Journal of Computer Vision*, vol. 88, pp. 303-338, 2010.

[7] Ghalleb A., Boumaiza S., and Amara N., "Demographic Face Profiling Based on Age, Gender and Race," *in Proceedings of 5th International Conference on Advanced Technologies for Signal and Image Processing*, Sousse, pp. 1-6, 2020.

[8] Gollapudi S., "Object Detection and Recognition," *in Proceedings of Learn Computer Vision Using OpenCV*, Berkeley, pp. 97-117, 2019.

[9] Girshick R., Donahue J., Darrell T., and Malik J., "Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation," *in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Columbus, pp. 580-587, 2014.

[10] Girshick R., "Fast R-Cnn," *in Proceedings of the IEEE International Conference on Computer Vision*, pp. 1440-1448, 2015.

[11] Felzenszwalb P., McAllester D., and Ramanan D., "A Discriminatively Trained, Multiscale, Deformable Part Model," *in Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, Anchorage, pp. 1-8, 2008.

[12] Huang M. and Wu Y., "GCS-YOLOV4-Tiny: A Lightweight Group Convolution Network for Multi-Stage Fruit Detection," *Mathematical Biosciences and Engineering*, vol. 20, no. 1, pp. 241-268, 2022.

[13] He K., Zhang X., Ren S., and Sun J., "Deep Residual Learning for Image Recognition," *in Proceedings IEEE Conference on Computer Vision Pattern Recognition,* Las Vegas, pp. 770-778, 2016.

[14] Jiang Z., Zhao L., Li S., and Jia Y., "Real-Time Object Detection Method for Embedded Devices," *Computer Vision and Pattern Recogniti*on, vol. 3, pp. 1-11, 2020.

[15] Jamiya S. and Rani E., "LittleYOLO-SPP: A Delicate Real-Time Vehicle Detection Algorithm," *Optik*, vol. 225, pp. 165818, 2021.

[16] Kessentini Y., Besbes M., Ammar S., and Chabbouh A., "A Two-Stage Deep Neural Network for Multi-Norm License Plate Detection and Recognition," *Expert Systems with Applications*, vol. 136, pp. 159-170, 2019.

[17] Kong W., Hong J., Jia M., Yao J., Cong W., Hu H., and Zhang H., "YOLOv3-DPFIN: A Dual-Path Feature Fusion Neural Network for Robust Real-Time Sonar Target Detection," *IEEE Sensors Journal*, vol. 20, no. 7, pp. 3745-3756, 2019.

[18] Lin Y., Cai R., Lin P., and Cheng S., "A Detection Approach for Bundled Log Ends Using K-Median Clustering and Improved Yolov4-Tiny

Network," *Computers and Electronics in Agriculture*, vol. 194, pp. 106700, 2022.

[19] Liu W., Anguelov D., Erhan D., Szegedy C., Reed S., Fu C., and Berg A., "Ssd: Single Shot Multibox Detector," *in Proceedings of European Conference on Computer Vision*, Amsterdam, pp. 21-37, 2016.

[20] Lin T., Maire M., Belongie S., Hays J., Perona P., Ramanan D., Dollár P., and Zitnick C., "Microsoft Coco: Common Objects Incontext," *in Proceedings of Computer Vision-ECCV*, Zurich, pp. 740-755, 2014.

[21] Mzoughi H., Njeh I., Wali A., Slima M., BenHamida A., Mhiri C., and Mahfoudhe K., "Deep Multi-Scale 3D Convolutional Neural Network (CNN) For MRI Gliomas Brain Tumor Classification," *Journal of Digital Imaging*, vol. 33, no. 4, pp. 903-915, 2020.

[22] Nasri M., Hmani M., Mtibaa A., Petrovska-Delacretaz D., Slima M., and Hamida A., "Face Emotion Recognition From Static Image Based on Convolution Neural Networks," *in Proceedings of 5th International Conference on Advanced Technologies for Signal and Image Processing*, Sousse, pp. 1-6, 2020.

[23] Niu J., Chen Y., Yu X., Li Z., and Gao H., "Data Augmentation on Defect Detection of Sanitary Ceramics," *in Proceedings of IECON the 46th Annual Conference of the IEEE Industrial Electronics Society*, Singapore, pp. 5317-5322, 2020.

[24] Ogundoyin S., "An Autonomous Lightweight Conditional Privacy-Preserving Authentication Scheme with Provable Security for Vehicular Ad-Hoc Networks," *International Journal of Computers and Applications*, vol. 42, no. 2, pp. 1-16, 2018.

[25] Pokkuluri K., Nedunuri S., "Crop Disease Prediction with Convolution Neural Network (CNN) Augmented with Cellular Automata," *The International Arab Journal of Information Technology*, vol. 19, no. 5, pp. 765-773, 2022.

[26] Felzenszwalb P., Girshick R., McAllester D., and Ramanan D., "Object Detection with Discriminatively Trained Part-Based Models," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 9, pp. 1627-1645, 2009.

[27] Prasetyo E., Suciati N., and Fatichah C., "Yolov4-Tiny and Spatial Pyramid Pooling for Detecting Head and Tail of Fish," *in Proceedings of International Conference on Artificial Intelligence and Computer Science Technology*, Yogyakarta, pp. 157-161, 2021.

[28] Piotrowski A. and Napiorkowski J., "A Comparison of Methods to Avoid Overfitting in Neural Networks Training in The Case of Catchment Runoff Modelling," *Journal of Hydrology*, vol. 476, pp. 97-111, 2013.

[29] Qi H., Xu T., Wang G., Cheng Y., and Chen C., MYOLOv3-Tiny: "A New Convolutional Neural Network Architecture for Real-Time Detection of Track Fasteners," *Computers in Industry*, vol. 123, pp. 103303, 2020.

[30] Ren S., He K., Girshick R., and Sun J., "Faster r-Cnn: Towards Real-Time Object Detection with Region Proposal Networks," *Advances in Neural Information Processing Systems*, vol. 28, 2015.

[31] Ren C., Kim D., and Jeong D., "A Survey of Deep Learning in Agriculture: Techniques and Their Applications," *Journal of Information Processing Systems*, vol. 16, no. 5, pp. 1015-1033, 2020.

[32] Redmon J., Divvala S., Girshick R., and Farhadi A., "You Only Look Once: Unified, Real-Time Object Detection," *in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Las Vegas, pp. 779-788, 2016.

[33] Redmon J. and Farhadi A, "YOLO9000: Better, Faster, Stronger," *in Proceedings of* IEEE *Conference on Computer Vision and Pattern Recognition*, Honolulu, pp. 6517-6525, 2017.

[34] Redmon, J., and Farhadi, A, "YOLOv3: An incremental improvement," 2018, arXiv:1804.02767. [Online]. Available: /web/20230216105855/https://arxiv.org/abs/1804.02767, Last Visited, 2023.

[35] Ranjbar M., Mori G., and Wang Y., "Optimizing Complex Loss Functions in Structured Prediction," *in Proceedings of European Conference on Computer Vision*, Heraklion, pp. 580-593, 2010.

[36] Redmon J., Darknet: Open source neural networks /web/20221224110653/https://pjreddie.com/darknet/, Last Visited, 2021.

[37] Srivastava N., Hinton G., Krizhevsky A., Sutskever I., and Salakhutdinov R., "Dropout:A Simple Way to Prevent Neural Networks From Overfitting," *The Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929-1958, 2014.

[38] Shao S., Zhao Z., Li B., Xiao T., Yu G., Zhang X., and Sun J., "Crowdhuman: A Benchmark for Detecting Human in A Crowd," *arXiv preprint arXiv:1805.00123*, 2018.

[39] Wang Y., Jia K., and Liu P., "Impolite Pedestrian Detection by Using Enhanced Yolov3-Tiny," *Journal of Artificial Intelligence*, vol. 2, no. 3, pp. 113-124, 2020.

[40] /web/20221224110042/https://www.anavid.co/, Last Visited, 2021.

[41] /web/20221224110203/https://github.com/Cartucho/mAP, Last Visited, 2021.

[42] /web/20221224110324/https://github.com/AlexeyAB/darknet, Last Visited, 2021.

[43] /web/20221224110506/http://host.robots.ox.ac.uk/pascal/VOC/, Last Visited, 2021.

[44] /web/20221224110807/https://cocodataset.org/

[45] /web/20221224111240/https://www.crowdhuman.org/, Last Visited, 2021.

[46] /web/20221224111209/https://www.cis.upenn.edu/~jshi/ped_html/, Last Visited, 2021.

[47] Wang L., Shi J., Song G., and Shen I., "Object Detection Combining Recognition and Segmentation," *in Proceedings of Asian Conference on Computer Vision*, Tokyo, pp. 189-199, 2007.

[48] Xun Z., Wang L., and Liu Y., "Improved Face Detection Algorithm Based on Multitask Convolutional Neural Network for Unmanned Aerial Vehicles View," *Journal of Electronic Imaging*, vol. 31, no. 6, pp. 061804, 2022.

[49] Yang Z., Xu W., Wang Z., He X., Yang F., and Yin Z., "Combining YOLOV3-Tiny Model with Dropblock for Tiny-Face Detection," *in Proceedings of IEEE 19th International Conference on Communication Technology*, Xi'an, pp. 1673-1677, 2019.

[50] Yolo: Open Source Neural Networks in C. Availableonline: /web/20221224105904/https://pjreddie.com/darknet/yolo/, Last Visited, 2021.

[51] Ying X., "An Overview of Overfitting and Its Solutions," *in Journal of Physics*: *Conference Series*, vol. 1168, no. 2, pp. 022022, 2019.

[52] Yi Z., Yongliang S., and Jun Z., "An Improved Tiny-Yolov3 Pedestrian Detection Algorithm," *Optik*, vol. 183, pp. 17-23, 2019.

[53] Zhang P., Zhong Y., and Li X., "SlimYOLOv3: Narrower, Faster And Better for Real-Time UAV Applications," *in Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*, Seoul, 2019.

[54] Zhang S., Wen L., Bian X., Lei Z., and Li S., "Single-shot Refinement Neural Network For Object Detection," *in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Salt Lake City, pp. 4203-4212, 2018.

Donia Ammous obtained her bachelor degree at FSS (Faculty of Sciences of Sfax), Tunisia, in 2008. She received her MS degree in Electrical Engineers from the National School of Engineering (ENIS), Sfax, Tunisia, in 2012. She is currently a PhD student in the Laboratory of Electronics and Information Technology (LETI) ENIS, University of Sfax. Her main research activities include image\video processing on H.264/AVC, lossless video compression, cryptography and data security, remote sensing, UAV, computer vision and deep learning.

Achraf Chabbouh received his engineering degree from Higher School of Communication of Tunis. He currently works as University Teacher at Higher Institute of Technological Studies of Sidi bouzid, Tunisia. He is an experienced team player with a strong technical background especially in artificial intelligence, web and mobile application technology. He coordinates multiple complexes IT projects with many stakeholders in different fields, such as retail, agriculture, geospatial and finance.

Awatef Edhib received her engineering degree National Engineering School of Sfax (ENIS) in 2018. She currently works as a Research and Development IA engineer for Sogimel. She is passionate about artificial intelligence and innovation. She has a strong technical background in artificial intelligence, especially deep learning and computer vision.

Ahmed Chaari Ahmed Chaari received his Ph.D. in Automation and Industrial Engineering from Lille University, France in 2009. He worked as IT Program Manager in different companies from 2010 to 2018 in France, Sweden and Portugal. He is currently General Manager at Anavid France. His research interests include artificial intelligence, computer vision and data analysis.

Fahmi Kammoun received the DEA degree in automatic and signal processing from the University of Pierre et Marie Curie (Paris VI)-France in 1987, the Ph.D. degree in signal processing from the University of Orsay (Paris XI)-France in 1991. His doctoral work focused on the luminance uniformity, the contrast enhancement, the edges detection and gray-level video analysis. He received the HDR degree in electrical engineering from Sfax National School of Engineering (ENIS)-Tunisia in 2007. He is currently a professor in the department of physics at the Faculty of Sciences of Sfax (FSS)-University of Sfax. He is a member of the Laboratory of Electronics and Information Technology (LETI) - Tunisia. His current research interests include video quality metrics, video compression, video encryption, face and silhouette recognition, and Artificial Intelligence.

Nouri Masmoudi received his electrical engineering degree from the Faculty of Sciences and Techniques-Sfax, Tunisia, in 1982, the DEA degree from the National Institute of Applied Sciences—Lyon and University Claude Bernard-Lyon, France in 1984. From 1986 to 1990, he achieved his Ph.D. degree at the National School Engineering of Tunis (ENIT), Tunisia and obtained in 1990. He is currently a professor at the electrical engineering department, ENIS. Since 2000, he has been a director of ‘Circuits and Systems’ in the Laboratory of Electronics and Information Technology. Since 2003, he has been responsible for the Electronic Master Program at ENIS. His research activities have been devoted to several topics: Design, Telecommunication, Embedded Systems, Information Technology, Video Coding and Image Processing.