

Pedestrian detection under weather conditions using conditional generative adversarial network

Mohammed Razzok¹, Abdelmajid Badri¹, Ilham EL Mourabit¹, Yassine Ruichek², Aïcha Sahel¹

¹Laboratory of Electronics, Energy, Automation, and Information Processing, Faculty of Sciences and Techniques Mohammedia, University Hassan II Casablanca, Mohammedia, Morocco

²Laboratory Connaissances et Intelligence Artificielle Distribuées (CIAD), University of Technology of Belfort-Montbéliard (UTBM), Belfort, France

Article Info

Article history:

Received Sep 12, 2022

Revised Jan 25, 2023

Accepted Jan 30, 2023

Keywords:

Generative adversarial network

Noise removable

Pedestrian detection

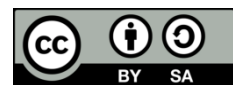
Weather conditions

You only look once

ABSTRACT

Nowadays, many pedestrians are injured or killed in traffic accidents. As a result, several artificial vision solutions based on pedestrian detection have been developed to assist drivers and reduce the number of accidents. Most pedestrian detection techniques work well on sunny days and provide accurate traffic data. However, detection decreases dramatically in rainy conditions. In this paper, a new pedestrian detection system (PDS) based on generative adversarial network (GAN) module and the real-time object detector you only look once (YOLO) v3 is proposed to mitigate adversarial weather attacks. Experimental evaluations performed on the VOC2014 dataset show that our proposed system performs better than models based on existing noise reduction methods in terms of accuracy for weather situations.

This is an open access article under the [CC BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) license.



Corresponding Author:

Mohammed Razzok

Laboratory of Electronics, Energy, Automation, and Information Processing

Faculty of Sciences and Techniques Mohammedia, University Hassan II Casablanca

Mohammedia, Morocco

Email: mohammedrazzok@gmail.com

1. INTRODUCTION

Road incidents are one of the most common causes of death worldwide today. As a result, pedestrian detection algorithms, which find all pedestrians in an image, have gained popularity in computer vision and artificial intelligence communities. Common pedestrian detection systems (PDS) are built for bright weather [1]–[9]. However, an applicable PDS is also required to perform well in rainy or snowy conditions.

Rain is one of the commonest dynamic weather phenomena. Images taken in rainy conditions frequently suffer from local degradations [10], [11] such as low visibility and distortion, which directly impair visual perception quality and make them unfit for sharing and use. Furthermore, rainwater-induced artifacts may drastically impact the performance of numerous machine vision solutions, such as smart driving and video monitoring systems.

Machine learning (ML) field focuses on the development of computer algorithms, which exploit data to learn patterns, make predictions, and increase their performance over time by more data. Lately, taking advantage of the invention of convolutional neural networks [12]–[15], particularly the establishment of the pix2pix [16] network architecture and the adversarial training strategy, the performance of single image de-raining has experienced notable progress. By training a rainy-to-clean image translation model with synthetic rain streak or raindrop datasets, a rainy image can be effectively repaired by eliminating the artifacts despite the presence of rain streaks or raindrops with different scales, forms, and thicknesses.

In this paper, we investigate the impact of images taken in rainy conditions on pedestrian classification tasks using mAP measurement. In addition, we assess the effectiveness of our proposed PDS based on Pix2Pix and you only look once (YOLO) v3, in comparison to others models based on noise-removing masks. The rest of this article is organized: Section 2 describes in detail the algorithms that will be employed. Section 3 introduces our proposed pedestrian detection system for combating adversarial weather attacks. In the next section, we present and discuss the results of our research. The last section addresses the paper's conclusion.

2. METHODS AND TOOLS

2.1. YOLO v3 algorithm

YOLO [17] is an open source object detection and classification algorithm based on convolutional neural networks (CNN). It is able to predict which objects are present in an image and their positions at first glance. The primary benefit of this approach is that the whole image is evaluated by a singular neural network. The network can process images in real-time at 45 frames a second (FPS) using Nvidia Titan X, and a simplified version fast YOLO can reach 155 FPS with better results compared than most real-time detectors.

YOLO starts detecting objects by dividing the input image into $S \times S$ grid, and each grid predicts C class probabilities, B bounding locations, and confidence scores. Each boundary box includes 5 variables: x , y , w , h , and a box confidence score. The confidence score represents how likely the bounding box includes an object and how precise the boundary box is. x and y are offsets to the corresponding cell. The bounding box width w and height h are normalized by the width and height of the image. Each cell has C conditional class probabilities. The final output of the YOLO has a shape of $(S, S, B \times 5 + C)$. The structure of the YOLO v3 algorithm is presented in Figure 1.

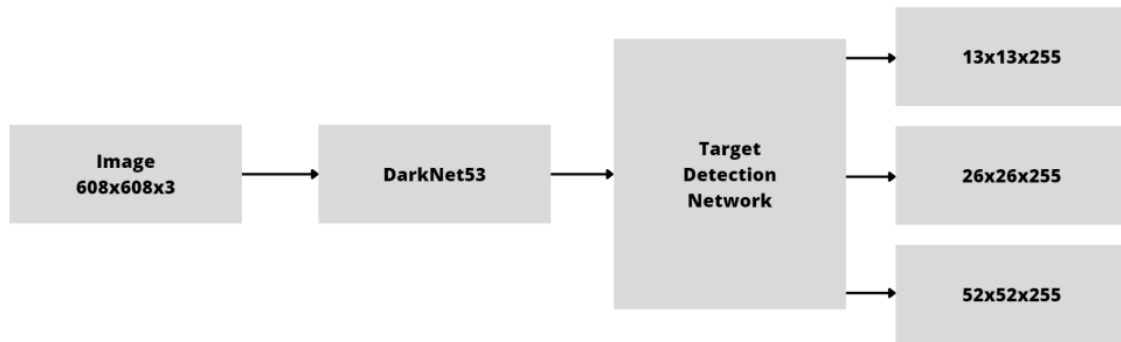


Figure 1. YOLO v3 architecture

2.2. Average filter

The average filter [18] operates by passing through the image pixel by pixel. At each location, the core element is replaced with the mean of the whole pixel values under the kernel region. The 3 by 3 and 5 by 5 filters are shown in (1) and (2) respectively.

$$K_3 = \frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \quad (1)$$

$$K_5 = \frac{1}{25} \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix} \quad (2)$$

2.3. Gaussian filter

The Gaussian filter [19] is a linear filter commonly used in image processing for blurring and removing details and noises from images. It has a different kernel that reflects the form of the Gaussian (bell-shaped) hump. An image I filtered by Gaussian convolution is given by (3), where σ is the standard deviation

of the distribution, p denotes the central pixel of the kernel, q represents the positions of its neighbors, and G_σ denotes the 2D Gaussian kernel (4).

$$GC[I_p] = \sum_{q \in S} G_\sigma(\|p - q\|) I_q \quad (3)$$

$$G_\sigma(x) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2}{2\sigma^2}} \quad (4)$$

The operation of the Gaussian convolution is not affected by the image content. The influence of one pixel on another in an image is defined only by their distance in the image, not by the image values themselves. The gaussian filters (kernel 3×3 , $\sigma=0.8$) and (kernel 5×5 , $\sigma=1.1$) are depicted in (5) and (6), respectively.

$$K_3 = \frac{1}{16} \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix} \quad (5)$$

$$K_5 = \frac{1}{256} \begin{bmatrix} 1 & 4 & 6 & 4 & 1 \\ 4 & 16 & 24 & 16 & 4 \\ 6 & 24 & 36 & 24 & 6 \\ 4 & 16 & 24 & 16 & 4 \\ 1 & 4 & 6 & 4 & 1 \end{bmatrix} \quad (6)$$

2.4. Median filter

The median filter [20] is a nonlinear filter that replaces the central pixel with the median of the pixels under the kernel area. The central element is always replaced by one of the pixels under kernel area. This is not the case with average and gaussian filtering. As a result, the median filter is less vulnerable to intense values (called outliers) than the average filter.

2.5. Bilateral filter

Bilateral Filter [21] is a technique for smoothing images and reducing noise without blurring large, sharp edges. It has the same definition as Gaussian convolution. On the other hand, it considers the value differences between neighbors. It's abbreviated as BF [,] and has the following definition,

$$BF[I_p] = \frac{1}{W_p} \sum_{q \in S} G_{\sigma_s}(\|p - q\|) G_{\sigma_r}(|I_p - I_q|) I_q \quad (7)$$

Where normalization factor W_p ensures pixel weights sum to 1.0:

$$W_p = \sum_{q \in S} G_{\sigma_s}(\|p - q\|) G_{\sigma_r}(|I_p - I_q|) \quad (8)$$

Parameters σ_s and σ_r will specify the amount of filtering for the image. G_{σ_s} is a spatial Gaussian weighting that decreases the influence of distant pixels, and G_{σ_r} is a range Gaussian that decreases the influence of pixels q when their intensity values differ from I_p . Bilateral filters are generated by `bilateralFilter(src, d, sigmaColor, sigmaSpace)` function [22], this function accepts the following parameters,

- `src`: the source image.
- `d`: the diameter of the pixel neighborhood.
- `sigmaColor`: the filter sigma in the color space.
- `sigmaSpace`: the filter sigma in the coordinate space.

2.6. Non local means filter

Non-local means method [23] fills the pixel's value with an average of the values of a distribution of other pixels: small blocks centered on other pixels are compared to the block centered on the pixel of interest, and the average is only conducted for pixels with blocks that are similar to the current block. As a consequence, this approach is capable of restoring textures that were previously blurred by other noising algorithms. Non Local Means filters are generated by `fastNlMeansDenoisingColored(src, h, hColor, templateWindowSize, searchWindowSize)` function [22], this function accepts the following parameters,

- `h`: A parameter that controls the filter strength for the luminance component. A larger `h` value removes all noise but also all image details; a smaller `h` value conserves details but also some noise.
- `hColor`: Identical to `h`, but it's for color images.

- `templateWindowSize`: The template block's size in pixels is used to calculate weights.
- `searchWindowSize`: The window size in pixels used to calculate the weighted mean for a particular pixel.

2.7. Adversarial network

Generative adversarial networks (GAN) [16] are a class of machine learning frameworks designed by Ian Goodfellow and al. GAN is composed of two parts: i) the generator learns to build real data. For the discriminator, the created instances serve as negative training examples, ii) the discriminator learns to make a distinction between fake and real data generated by the generator. When the generator makes implausible results, it is punished by the discriminator.

When training starts, the generator produces fake data, which the discriminator quickly recognizes. As training advances, the generator comes closer to generating output that fools the discriminator as training progresses. Finally, if generator training is efficient, the discriminator becomes less capable of distinguishing between real and fabricated images. It begins to mistakenly classify fake data as real, and its accuracy decreases as a result.

Both models are based on neural networks. The discriminator input is connected to the generator output directly. The discriminator's classification is used by the generator to update its weights via back-propagation. As a result, both models have been trained concurrently in an adversarial process in which the generator tries to trick the discriminator while the discriminator attempts to spot the fake pictures. The GAN framework is presented in Figure 2.

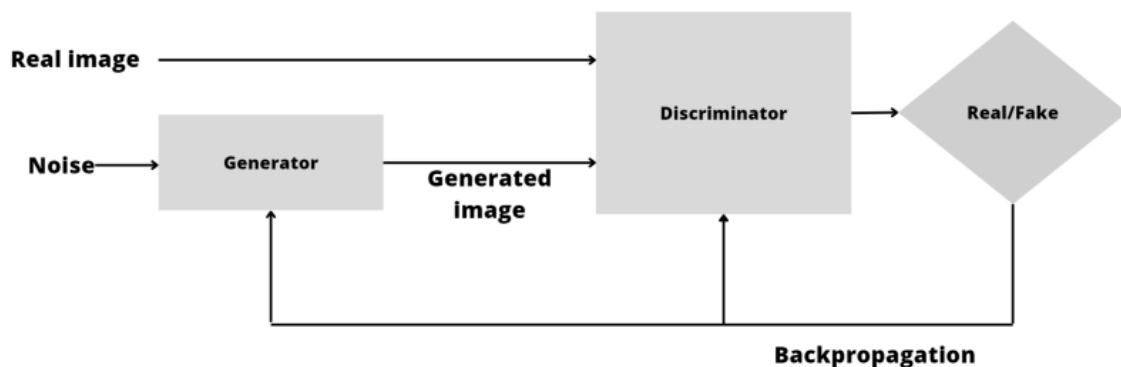


Figure 2. GAN architecture

3. PROPOSED METHOD

To eliminate the rainy effect from images, we attempt to directly turn a rainy image into an unrainy image pixel by pixel directly. This study was inspired by the effectiveness of Pix2Pix GANs in translating one image into another. The Pix2Pix technique uses a conditional GAN (cGAN), in which the output picture is produced in response to an input, in our particular scenario a source image.

The discriminator is a type of image classification model that uses a Deep CNN. Specifically, it performs conditional-image classification by taking both the source image (e.g. a rainy image) and the target image (e.g. an unrainy image) as input, and then predicts the probability of whether the target image is real or a fake version of the source image. The PatchGAN model, which is based on the efficiency of the model's receptive field, is used to define the relationship between one of the model's outputs and the number of pixels in the input image. This model is designed so that each output prediction maps to a 70×70 block of the input image. The advantage of using this model is that it can handle images of different sizes, such as those larger or smaller than 256×256 pixels. During training, the model generates a patch of predictions by concatenating two input images. To optimize the model, it uses log loss and applies a weighting factor of 0.5 to updates, which is a technique recommended by the Pix2Pix authors. This weighting slows down changes to the discriminator model compared to the generator model, which helps improve the overall training process. The flowchart of our proposed discriminator is presented in Figure 3.

In comparison to the discriminator, the generator is more complicated. The generator employs a U-Net architecture as an encoder-decoder model. It generates a target image (unrainy image) from a source image (rainy image). To achieve this, the input image is first downscaled or encoded to a bottleneck layer, and then the condensed representation is upsampled or decoded to the output image size. Figures 4 to 6 depict the flowcharts for the encoder, decoder, and generator respectively.

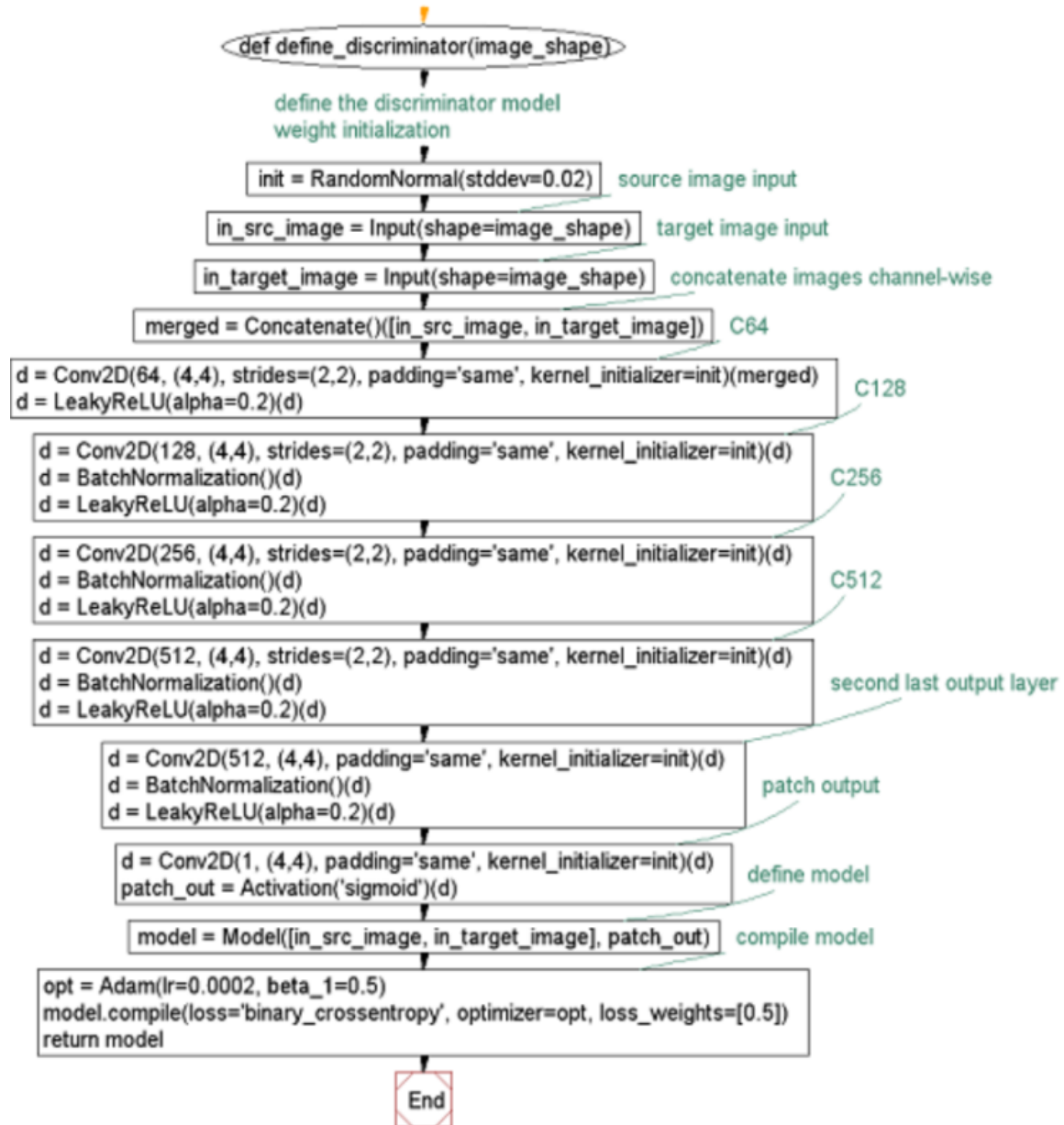


Figure 3. The flowchart of the discriminator algorithm

The discriminator model is trained particularly on both real and fake images, whereas the generator model is not. Furthermore, it is trained using the discriminator and updated to reduce the discriminator's predicted loss for "real" generated images. In this way, it is encouraged to produce more realistic images. The weights are reemployed in this composite model, but they are identified as untrainable because the discriminator is updated separately. The composite model is updated with two targets: one confirming that the produced images were authentic (cross-entropy loss), which forces the generator to make large weight updates to produce more realistic images, and the actual real translation of the image, which is compared to the generator model's output (L1 loss).

GAN models hardly ever converge. Instead, a balance is established between the generator and discriminator models. As a result, it is difficult to decide when to stop training. During training, we can save the model regularly and use it to generate sample image-to-image translations. For example, after 10 training epochs, we examine the generated images and choose a final model based on the image quality. Figure 7 depicts the Pix2Pix cGAN flowchart.

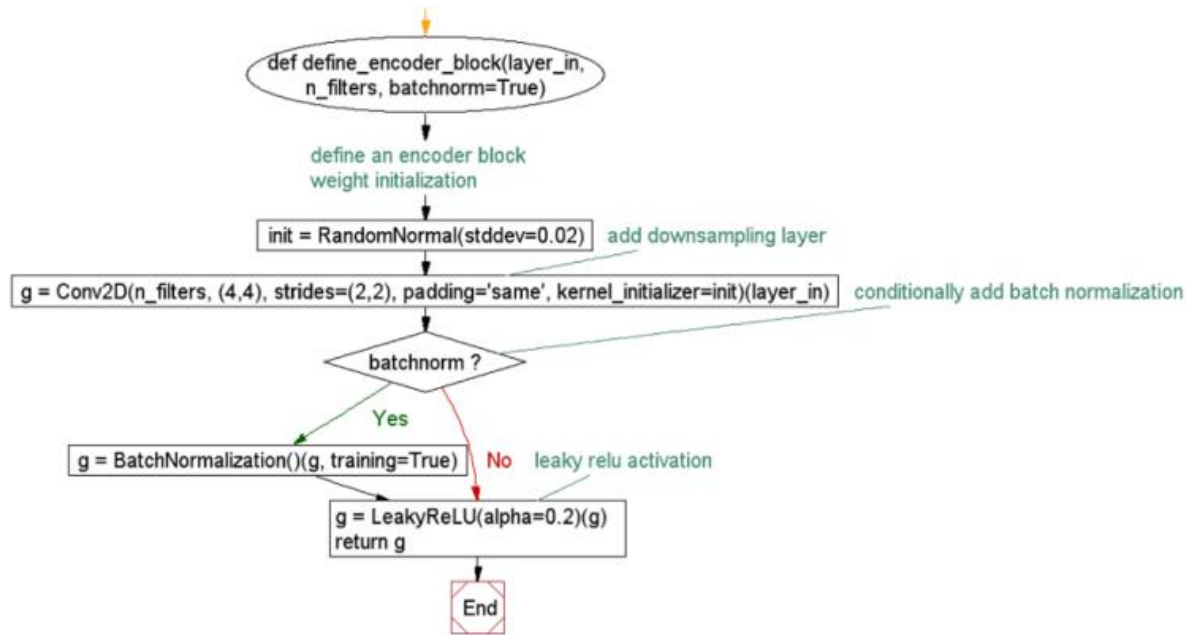


Figure 4. The flowchart of the encoder model

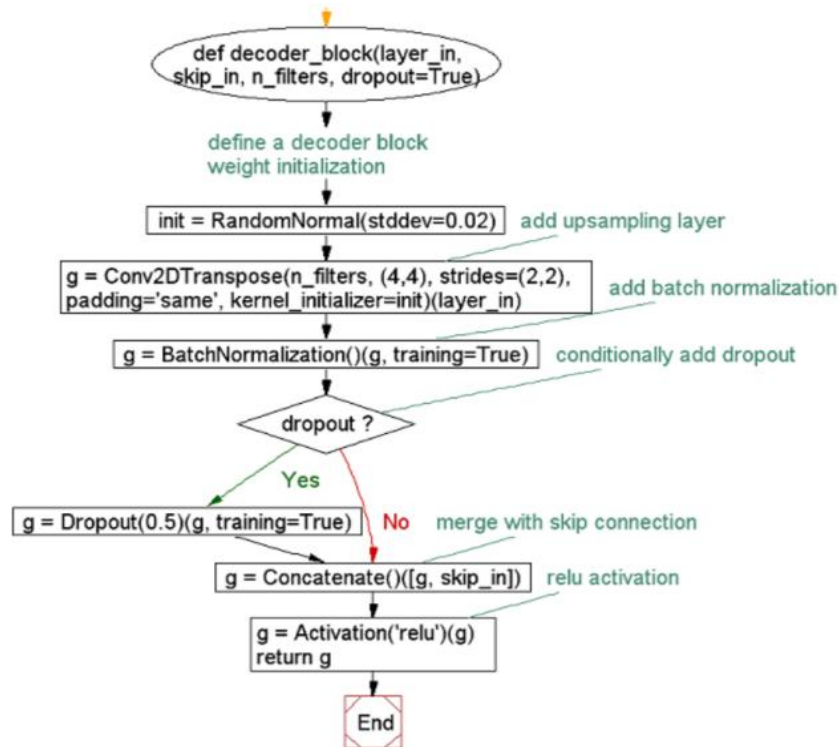


Figure 5. The flowchart of the decoder model

In this study, we proposed a pedestrian detection system capable of detecting pedestrians in two scenarios rainy and un-rainy as presented in Figure 8. The first step of our system is to determine whether it is raining on the input images or not using our proposed rainy detector based on deep convolutional neural networks as described in Figure 9. If the image it's rainy we use our generator from the generative adversarial network to transform the rainy image to un-rainy image, if not it passes directly to YOLO v3 to detect the

positions of pedestrians in the image. Moreover, to demonstrate the effectiveness of our proposed PDS model against weather degradation, we compared it with multiple PDS models based on removable noise filters presented in Figures 10 to 14.

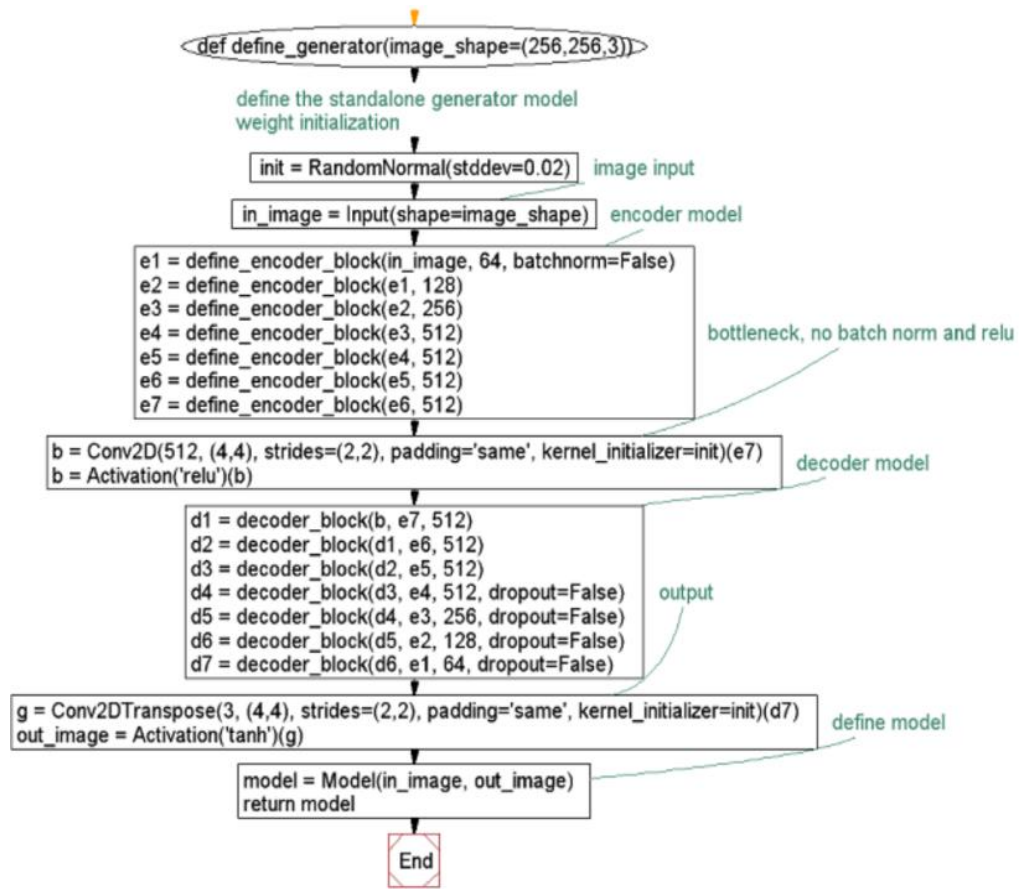


Figure 6. The flowchart of the generator model

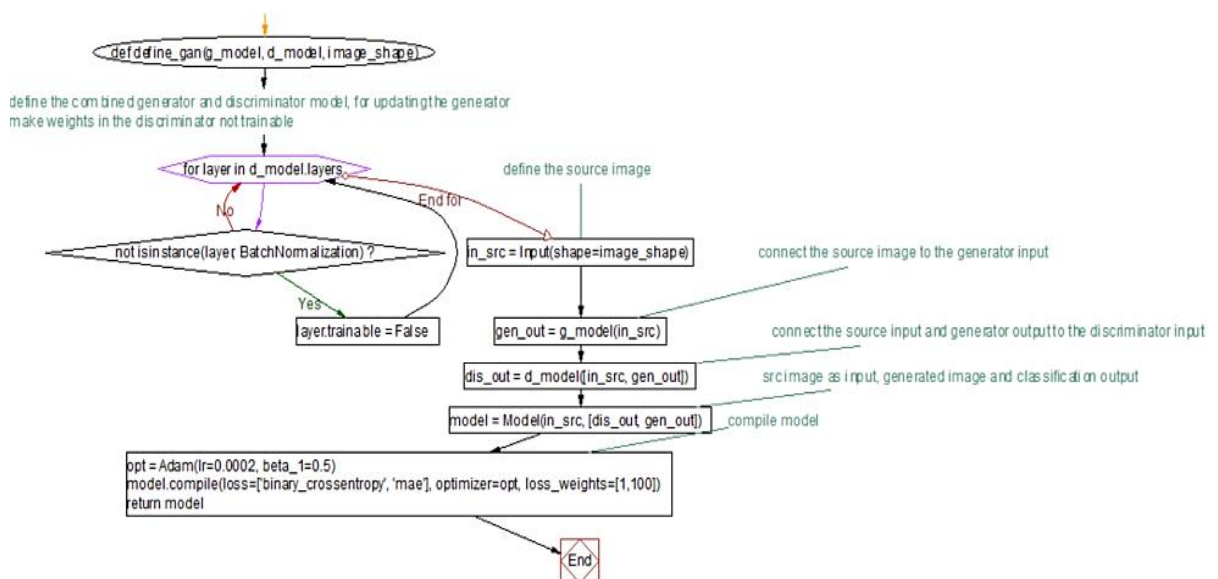


Figure 7. The flowchart of the Pix2Pix cGAN model

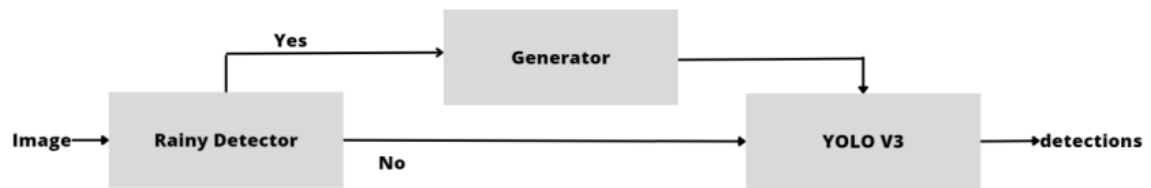


Figure 8. Proposed PDS based on GAN

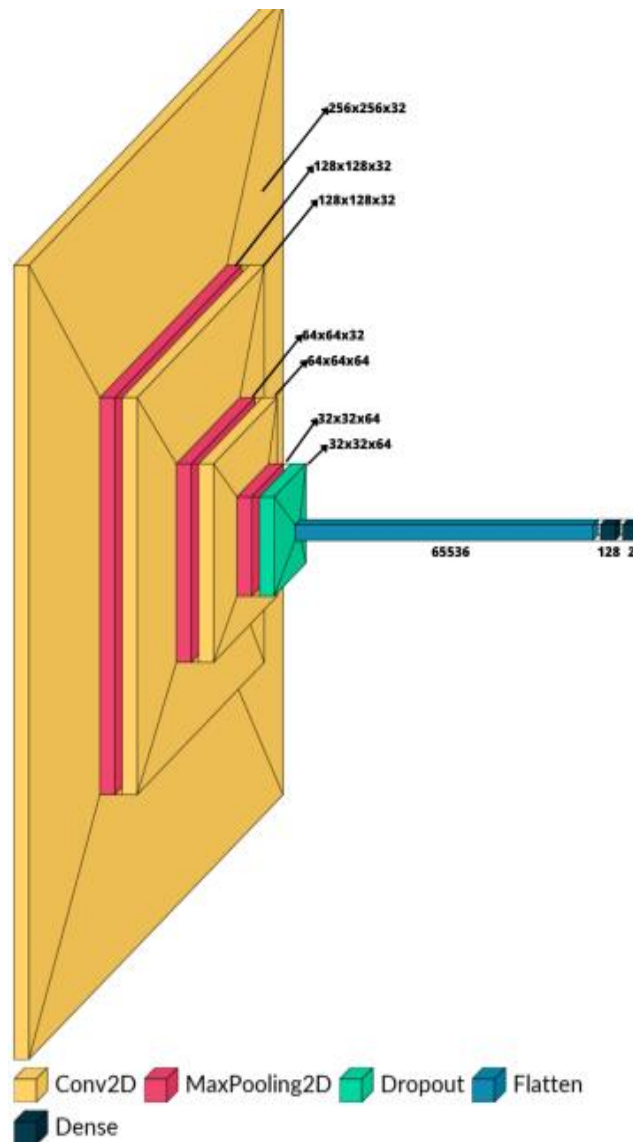


Figure 9. Rainy Detector Architecture

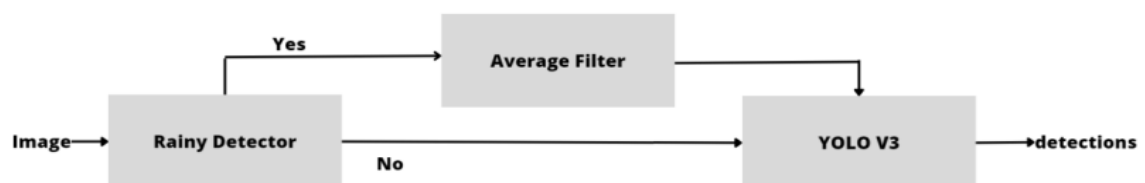


Figure 10. PDS based on average filter

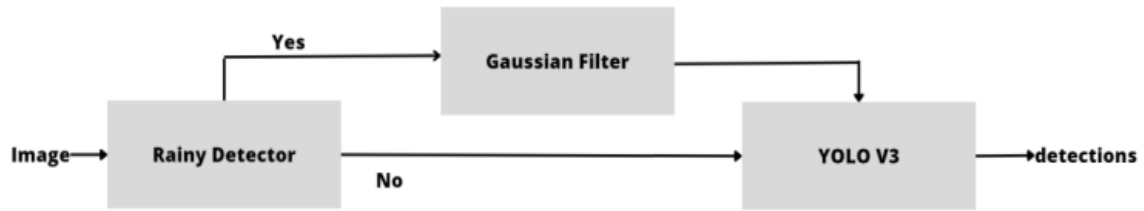


Figure 11. PDS based on Gaussian filter

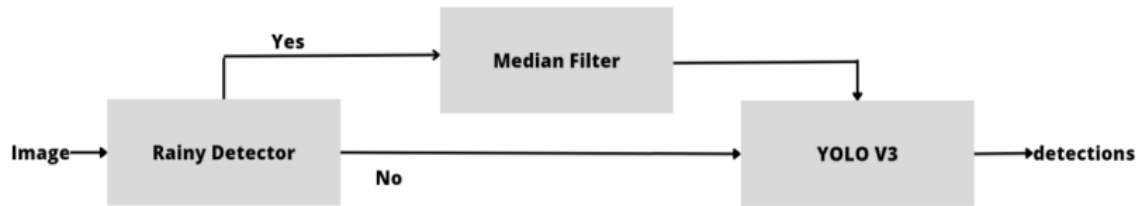


Figure 12. PDS based on median filter

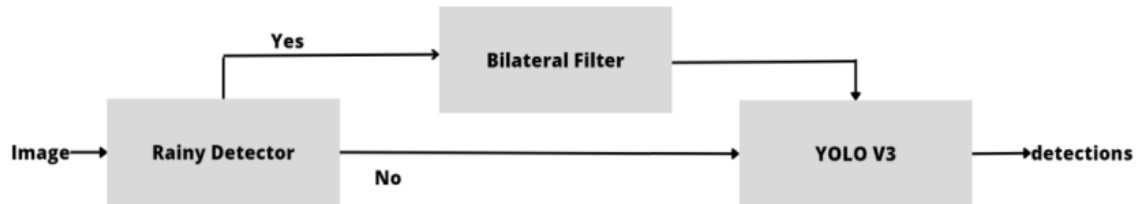


Figure 13. PDS based on bilateral filter

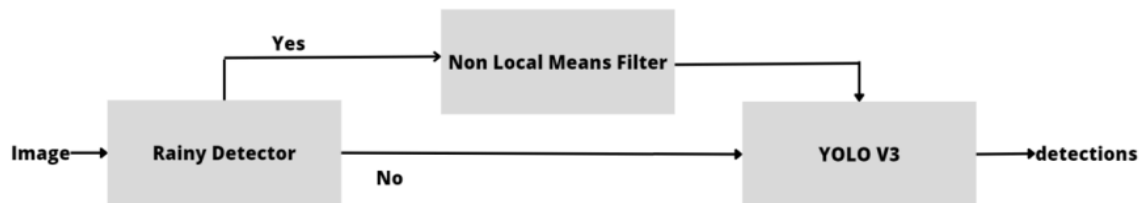


Figure 14. PDS based on non-local means filter

4. EXPERIMENTAL RESULTS & DISCUSSION

For the purpose of building our GAN model, we downsized the VOC2014 dataset [24] to 256×256 and divided it into two folders, one for training and one for testing. The train folder contains 1,000 images from number 000001 to 001979, whereas the testing dataset contains 3,952 images from number 001983 to 009963. Each image includes a pair of rainy images on the left and unrainy image on the right. Rainy images are generated using add_rain function from Automold [source code](#) (add_rain (clean, slant=-20, drop_length=20, drop_width=1, rain_type='heavy')). Additionally, we prepared our proposed Rainy detector CNN using 600 images (from 000001 to 001190) for training and 400 images (from 001193 to 001979) for validation.

All modules were implemented on Ubuntu 20.04 LTS with Python 3.7.13. Deep-learning networks were also implemented using the Tensorflow framework (version 2.6). In addition, YOLOv3 weights were initialized using the “yolov3.weights” COCO pre-trained model (416×416). Moreover, all experiments were performed using an Nvidia 1650 GPU 4GB, and an Intel(R) Core(TM) i5-9400F CPU 2.90GHz (6Cores) and 16GB RAM.

In our work we:

- Load and prepare the rainy-affected images from the original image dataset.
- Develop a Rainy Detector model to determine whether or not it is raining on the input images.

- Build a Pix2Pix model to transform rainy images to un-rainy images.
- Use the final Pix2Pix generator model to transform rainy images to un-rainy images.
- Use the pre-trained YOLO v3 model to detect pedestrians in images.

The average precision metric (AP), which measures the region below the precision-recall graph, was used to evaluate our models. It is a widely used metric for evaluating the accuracy of object detectors. We determined the average precision in this work using the Cartucho source code [25]. The results presented in Table 1 show that the average, Gaussian, median, bilateral, and local means filters do not help the images with raindrops, and instead make the detection result worse. On the other hand, our proposed system succeeded in restoring rainy images to un-rainy images and achieved better pedestrian detection performance. The only limitation of our work is that our system can achieve about 10 FPS for non rainy images and 6 FPS for rainy images due to the hardware and software limitations.

Table 1. Performance of proposed pedestrian detection systems

Model	Dataset			
	VOC2014		Rainy VOC2014	
	AP(%)	FPS(Hz)	AP(%)	FPS(Hz)
YOLOv3	74.38	14	26.5	14
Average filter(3,3) & YOLOv3	73.84	10	23.34	10
Gaussian filter(3,3) & YOLOv3	73.88	10	24.32	10
Median filter(3,3) & YOLOv3	73.89	10	25.2	10
Bilateral filter(3,10,10) & YOLOv3	73.95	10	26.09	10
Non Local Means Filter(3,3,3,3) & YOLOv3	73.94	10	25.47	9
Pix2Pix GAN rainy removal & YOLOv3	73.74	10	46.10	6

5. CONCLUSION

Advanced driving assistance systems are becoming more advanced with in-vehicle infrastructures. However, on rainy days, the detection rate remains poor. Rain streaks accumulate and obstruct the camera's view. In addition, most pedestrians wear raincoats or hold umbrellas on rainy days, resulting in a high number of occlusions. Due to the difficulty in detecting pedestrians in the rain, this study proposed a new PDS that includes a de-raining subsystem to detect pedestrians in both rainy and non-rainy conditions. Our proposed PDS outperforms both the existing YOLOv3 method and the traditional basic noises removable algorithms. Developing a neural network that excels in one area but fails in others is not a viable strategy for self-driving vehicles. Our long-term goal is to develop deep-learning architectures and solutions that can detect objects in a variety of environments. We are also interested in using thermal imaging cameras to detect pedestrians because of their ability to see in complete darkness, light fog, light rain, and snow.

ACKNOWLEDGEMENTS

This research work was supported by the University Hassan II Casablanca's Technology of Information and Communication Center as part of the Big Data & Connected Objects research project.




REFERENCES

- [1] M. Razzok, A. Badri, Y. Ruichek, and A. Sahel, "Street crossing pedestrian detection system a comparative study of descriptor and classification methods," *Colloque sur les Objets et systèmes Connectés*, 2019.
- [2] J. Mao, T. Xiao, Y. Jiang, and Z. Cao, "What can help pedestrian detection?," vol. 21, no. 1, pp. 1–9, May 2017, doi: 10.48550/arXiv.1705.02757.
- [3] W. Ouyang and X. Wang, "Joint deep learning for pedestrian detection," *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2056–2063, 2013, doi: 10.1109/ICCV.2013.257.
- [4] W. Lan, J. Dang, Y. Wang, and S. Wang, "Pedestrian detection based on yolo network model," *Proceedings of 2018 IEEE International Conference on Mechatronics and Automation, ICMA 2018*, pp. 1547–1551, 2018, doi: 10.1109/ICMA.2018.8484698.
- [5] J. Cao, Y. Pang, and X. Li, "Learning multilayer channel features for pedestrian detection," *IEEE Transactions on Image Processing*, vol. 26, no. 7, pp. 3210–3220, 2017, doi: 10.1109/TIP.2017.2694224.
- [6] C. Ning, L. Menglu, Y. Hao, S. Xueping, and L. Yunhong, "Survey of pedestrian detection with occlusion," *Complex and Intelligent Systems*, vol. 7, no. 1, pp. 577–587, 2021, doi: 10.1007/s40747-020-00206-8.
- [7] Y. Shao, X. Zhang, H. Chu, X. Zhang, D. Zhang, and Y. Rao, "AIR-YOLOv3: Aerial infrared pedestrian detection via an improved YOLOv3 with network pruning," *Applied Sciences (Switzerland)*, vol. 12, no. 7, 2022, doi: 10.3390/app12073627.
- [8] I. P. Alonso *et al.*, "Combination of feature extraction methods for SVM pedestrian detection," *IEEE Transactions on Intelligent Transportation Systems*, vol. 8, no. 2, pp. 292–307, 2007, doi: 10.1109/TITS.2007.894194.
- [9] S. Nedeveschi, S. Bota, and C. Tomiuc, "Stereo-based pedestrian detection for collision-avoidance applications," *IEEE Transactions on Intelligent Transportation Systems*, vol. 10, no. 3, pp. 380–391, 2009, doi: 10.1109/TITS.2008.2012373.




- [10] X. Zhi, S. Jiang, W. Zhang, D. Wang, and Y. Li, "Image degradation characteristics and restoration based on regularization for diffractive imaging," *Infrared Physics and Technology*, vol. 86, pp. 226–238, 2017, doi: 10.1016/j.infrared.2017.09.014.
- [11] M. Razzok, A. Badri, I. E. L. Mourabit, Y. Ruichek, and A. Sahel, "A new pedestrian recognition system based on edge detection and different census transform features under weather conditions," *IAES International Journal of Artificial Intelligence*, vol. 11, no. 2, pp. 582–592, 2022, doi: 10.11591/ijai.v11.i2.pp582-592.
- [12] J. Gu *et al.*, "Recent advances in convolutional neural networks," *Pattern Recognition*, vol. 77, pp. 354–377, 2018, doi: 10.1016/j.patcog.2017.10.013.
- [13] A. Khan, A. Sohail, U. Zahoor, and A. S. Qureshi, "A survey of the recent architectures of deep convolutional neural networks," *Artificial Intelligence Review*, vol. 53, no. 8, pp. 5455–5516, 2020, doi: 10.1007/s10462-020-09825-6.
- [14] A. Dhillon and G. K. Verma, "Convolutional neural network: A review of models, methodologies and applications to object detection," *Progress in Artificial Intelligence*, vol. 9, no. 2, pp. 85–112, 2020, doi: 10.1007/s13748-019-00203-0.
- [15] M. Razzok, A. Badri, I. EL Mourabit, Y. Ruichek, and A. Sahel, "Pedestrian detection system based on deep learning," *International Journal of Advances in Applied Sciences*, vol. 11, no. 3, p. 194, 2022, doi: 10.11591/ijaas.v11.i3.pp194-198.
- [16] P. Isola, J. Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, vol. 2017-January, pp. 5967–5976, 2017, doi: 10.1109/CVPR.2017.632.
- [17] J. Redmon and A. Farhadi, "YOLOv3: An incremental improvement," 2018, [Online]. Available: <http://arxiv.org/abs/1804.02767>.
- [18] X. X. Sun and W. Qu, "Comparison between mean filter and median filter algorithm in image denoising field," *Applied Mechanics and Materials*, vol. 644–650, pp. 4112–4116, 2014, doi: 10.4028/www.scientific.net/AMM.644-650.4112.
- [19] B. K. Shreyamsha Kumar, "Image denoising based on Gaussian/bilateral filter and its method noise thresholding," *Signal, Image and Video Processing*, vol. 7, no. 6, pp. 1159–1172, 2013, doi: 10.1007/s11760-012-0372-7.
- [20] K. O. Boateng, B. W. Asubam, and D. S. Laar, "Improving the effectiveness of the median filter," *International Journal of Electronics and Communication Engineering*, vol. 5, no. 1, pp. 85–97, 2012, [Online]. Available: <http://www.irphouse.com>.
- [21] S. Paris, P. Kornprobst, J. Tumblin, and F. Durand, "Bilateral filtering: Theory and applications," *Foundations and Trends in Computer Graphics and Vision*, vol. 4, no. 1, pp. 1–73, 2009, doi: 10.1561/06000000020.
- [22] A. Zelinsky, "Learning OpenCV-computer vision with the OpenCV library," *IEEE Robotics and Automation Magazine*, vol. 16, no. 3, p. 100, 2009, doi: 10.1109/MRA.2009.933612.
- [23] B. K. Shreyamsha Kumar, "Image denoising based on non-local means filter and its method noise thresholding," *Signal, Image and Video Processing*, vol. 7, no. 6, pp. 1211–1227, 2013, doi: 10.1007/s11760-012-0389-y.
- [24] M. Everingham, S. M. A. Eslami, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes challenge: A retrospective," *International Journal of Computer Vision*, vol. 111, no. 1, pp. 98–136, 2015, doi: 10.1007/s11263-014-0733-5.
- [25] J. Cartucho, R. Ventura, and M. Veloso, "Robust object recognition through symbiotic deep learning in mobile robots," *IEEE International Conference on Intelligent Robots and Systems*, pp. 2336–2341, 2018, doi: 10.1109/IROS.2018.8594067.

BIOGRAPHIES OF AUTHORS






Mohammed Razzok    is received the Engineering degree in networks and telecommunications from the National School of Applied Sciences (ENSA), Oujda, Morocco, in 2017. He is currently pursuing his Ph.D. in the computer vision field at the Laboratory of Electronics, Energy, Automation and Information Processing, Faculty of Sciences and Techniques Mohammedia, University Hassan II Casablanca, Morocco. His main research is constructing and implementing on-board systems for intelligent vehicles to perform pedestrian recognition/detection, analyze pedestrian trajectories, warn the driver or initiate appropriate protective measures when detecting potentially dangerous situations. He can be contacted at email: mohammedrazzok@gmail.com.






Abdelmajid Badri    holds a Ph.D. in Electronics and Image Processing from the University of Poitiers, France, in 1992. In 1996, he got the diploma of authorization to manage researches at the University of Poitiers France. He is a director at the Higher School of Technology (EST), Casablanca, and he is a University Professor (PES-C) at FST Mohammedia, University Hassan II of Casablanca, Morocco where he teaches electronics, signal processing, image processing, and telecommunication. He is a member of the electronic, energy, automatic and information processing (EEA and TI) laboratory which he managed since 1996. His research focuses on communication and information technology (electronics systems, signal/image processing, and telecommunication). He is qualified by the CNU-France in the 61st section: computer engineering, automatic, and signal processing. He managed multiple doctoral theses. He is a co-author of several national and international publications and is responsible for many research projects financed by the ministry or by the industrialists. He was a member of several committees and programs of international conferences and president of three international congresses in the same domain. He is a member and co-responsible for multiple scientific associations in touch with his domain of research. He can be contacted at email: abdelmajid_badri@yahooo.fr.






Ilham EL Mourabit    holds a Ph.D. degree in Electronic Systems and Telecommunications in 2018. She is a University Professor at FST Mohammedia. She received the B.Sc. degree in Electronics and Computer Science, from the Faculty of Sciences and Technology of Mohammedia, Hassan II University Casablanca, Morocco, in 2009, and an M.Sc. degree in Electronic and Automatic Systems Engineering (telecommunication and information technologies specialty) from the same institution. In her research, she focused on how to use satellite and cellular network's signals to determine the accurate position of a mobile target to perform real-time monitoring. She can be contacted at email: elmourabit.ilham@gmail.com.



Yassine Ruichek    (Senior Member, IEEE) holds a Ph.D. degree in control and computer engineering in 1997 and the authorization to manage researches (HDR) degree in physic science from the University of Lille, France, in 2005. In 2007 he became a full professor at the University of Technology of Belfort-Montbéliard (UTBM). His research interests include computer vision, image processing and analysis, pattern recognition, data fusion, and localization, with applications in intelligent transportation systems and video surveillance. He can be contacted at email: yassine.ruichek@utbm.fr.



Aïcha Sahel    holds a Ph.D. in Electronics and Image Processing from the University of Poitiers, France, in 1996. She is a Professor at FST Mohammedia, University Hassan II of Casablanca, Morocco, and a member of the EEA and TI laboratory. Aïcha Sahel's research focuses on electronic systems, signal/image processing, and telecommunications. She co-supervises doctoral theses and co-authors of several national and international publications and is a member of multiple financed research projects. She was a member of the steering committees of three international congresses in the same field of research. She can be contacted at email: sahel_ai@yahoo.fr.