International Journal of Reconfigurable and Embedded Systems (IJRES)
Vol. 13, No. 3, November 2024, pp. 604~615
ISSN: 2089-4864, DOI: 10.11591/ijres.v13.i3.pp604-615

Moving objects detection based on histogram of oriented gradient algorithm chip for hazy environment

Monika Sharma[1], Kuldeep Singh Kaswan[1], Dileep Kumar Yadav[2]
[1]Department of Computer Science and Engineering, School of Computer Science and Engineering, Galgotia's University, Greater Noida, India
[2]Department of Computer Science and Engineering, School of Computer Science Engineering and Technology, Bennett University, Greater Noida, India

Article Info

Article history:

Received Aug 23, 2023
Revised Dec 25, 2023
Accepted Feb 2, 2024

Keywords:

FPGA synthesis
Hardware description language
HoG chip design
Object detection
Object tracking
Xilinx ISE 14.7

ABSTRACT

The most important aspects of computer vision are moving object detection (MOD) and tracking. Many signal-processing applications use regional image statistics. Compute-intensive video and image processing with low latency and high throughput is done with field programmable gate array (FPGA) image processing. Local image statistics are used for edge identification and filtering. The histogram of oriented gradients (HoG) algorithm extracts local shape characteristics by equalizing histograms. The objective of the work is to design the hardware chip of the algorithm and perform the simulation in the Xilinx ISE 14.7 simulation environment. The performance of the chip is evaluated in Modelsim 10.0 simulation software to check its feasibility. The performance of the chip design is estimated on Viretx-5 FPGA and compared with the MATLAB-2020 image processing tool-based response time. This form of tracking typically deals with identifying, anchoring, and tracking images and videos. A mask made from a cut-out of the object can then determine the plane's coordinates depending on its position. This type of object tracking is frequently utilized in the field of augmented reality (AR). The algorithm is most suited for object detection using hardware controllers in haze and foggy environments.

Corresponding Author:

Monika Sharma
Department of Computer Science and Engineering, School of Computer Science and Engineering
Galgotia's University
Greater Noida, Uttar Pradesh 203201, India
Email: monikasharma.cse@gmail.com

1. INTRODUCTION

Video target tracking includes [1] the identification of moving targets, which is crucial. Effective video track requires good moving target detection. The technique of extracting moving objects from image sequences that are relatively obvious to the eye from the backdrop based on their features such as intensity, edge, texture, and so forth is known as moving target detection. Its goal is to identify and separate stationary background targets from moving foreground targets. Simply said, moving target recognition is applied to establish the location of a moving object and determine whether it has been spotted in a video series.

There have been numerous moving target identification techniques, but the optical flow process, inter-frame differential method, background subtraction (BS), and various enhanced algorithms derived from them are the most significant. The optical flow approach is the most computationally intensive and has the highest hardware requirements, making it more challenging to attain the method's objective of real-time detection [2]. For background modeling and background updating, the BS method is substantially more labor-intensive. To avoid being influenced by environmental changes like illumination and interferences, we must swiftly build the backdrop image and refresh the background in time.

Modern intelligent systems make significant use of computer vision and video analysis tools [3]. Because cameras are simple to set up, use, and maintain, video-based systems may obtain a wider variability of needed data and are relatively economical. There is an imperative demand for automatic video-recognizing systems that can substitute human operatives to monitor the regions under surveillance given the vast number of video cameras that are put everywhere these days. Every object in intelligent systems that relies on video can be found and monitored by a good tracking system. A well-separated feature-based event detection model can be constructed after tracking results have been collected.

The five types of object-tracking techniques include appearance-based, model-based, feature-based, mesh-based, contour-based, and hybrid techniques [4]. The model-created tracking techniques take advantage of the geometry of typical items in a scene that is known in advance. The issue of tracking partially obscured objects can be resolved thanks to the construction of parameterized object models. The dynamic model of video things is used to track associated sections that roughly match the 2-D shapes of the video objects using appearance-based methods. The tracking approach depends on data that the complete region provides. This information can include things like movement, colour, and texture. Complex deformation typically exceeds the capability of these methods. Contour-based approaches just follow the outline of an object rather than each pixel. To project the contour and subsequently modify it to the object observed in the following frame. Rather than using static photos like in traditional object detection, video object detection uses video data to find objects. Video surveillance and autonomous driving are two applications that have significantly influenced the development of video object detection [5]. The ImageNet large scale visual recognition challenge (ILSVRC2015) added video target identification as a new task in 2015. ILSVRC2015 has contributed to an increase in research on video object detection.

Robotics and human-computer interaction are only two examples of the many industries that have used computer vision as a key application of smart embedded systems [6]. Unmanned vehicles, computerized traffic control, surveillance, living biological image analysis, and smart intelligent robots are just a few examples of applications where object tracking, a core part of computer vision, can be highly helpful. With the use of object tracking, moving objects in a video frame sequence can be tracked along their paths. Object tracking [7] requires intense processing to extract the needed information from large amounts of video input, like most computer vision jobs. High-speed object tracking methods are also required due to the real-time handling demands of certain computer vision and related applications [8]. Field programmable gate arrays (FPGAs) have emerged as desirable computation programs for complicated applications [9] for high-device performance and less power consumption demands as shrinking process advances have made it possible to fit more transistors onto a single silicon chip. They offer great adaptability for porting programs to spatially parallel architectures due to their large number of programmable logic blocks, large number of memory modules, and high-performance digital signal processing (DSP) components [10].

2. RELATED WORK

Images of outdoor sceneries frequently show fog, haze, mist, and other atmospheric degradation elements because air particles absorb light, which is then reflected by the source [11]. This effect influences how people see remote-sensing images. Histogram equalization, phase function consistency testing, and bilateral filters are all methods that utilize multi-retinex theory to reduce undesirable artifacts and improve the clarity of the result's visual appearance. The work concentrated on the suggested detection-based trailing system for fractures in ship assessment videos [12]. Using the ideal anchor programming and a postprocessing approach to get rid of terminated estimates, a customized RetinaNet model performs the detection stage. The enhanced channel and spatial reliability tracking (CSRT) and the novel data association algorithm are the two main parts of the tracking stage, which also maintains tracking indications for each trailing trajectory. The improved CSRT tracker expects the tracking information in the subsequent outlines by supporting an initial trailing target, and the innovative data relationship algorithm networks discoveries with the prevailing trackers.

The two primary responsibilities of video surveillance systems [13] are BS and moving object detection (MOD). The existence of noise in the video sequence that was taken, however, is one of the main issues that seriously compromises the accuracy of detection. In that work, authors worked on the new MOD approach called De-Noising and moving object processing by lower-ranked approximation method from noisy video data. The suggested method produces accurate visual findings and measures values. The suggested solution was evaluated under many testing conditions, including shadow, inclement weather, camera jitter, and dynamic background. The original algorithm improves transmittance and maximizes distinctive light value [14] while resolving the issue that the dark channel previous technique causes the image colour to degrade. The image was switched from red, green, and blue (RGB) to hue saturation value

(HSV) space based on the restoration method for further improvement. The multiscale retinex with colour restoration (MSRCR) technique was used to improve the V component throughout the process of enhancement, and the adaptive stretching strategy was used to improve the saturation. The simulation experiment findings demonstrate that when the traditional enhancement algorithm improves the image, the new approach addresses the drawbacks of noise amplification and edge blur. The authors proposed an innovative haze/fog removal method that splits a foggy image into high and low-frequency groups according to their operational information using tetrolet transformation and uses a residual frequency extractor based on dual dictionary learning to extract more residual image data [15]. Sharpening the tetrolet coefficients extracts more precise information while performing dark channel prior (DCP) operation on the lower-frequency section to improve more fog-free information. When the inverse converted image is combined with the remaining high-frequency image component, contrast-constrained adaptive histogram equalization is used in post-processing to equalize the balance of contrast.

An efficient method for removing haze from images is suggested by the authors, and it is based on multiexposure image fusion and better colour channel transfer [16]. A colour channel transport procedure based on k-means methods is used as part of the initial preprocessing of the image. A series of multiexposure images are then created using gamma correction, which is introduced based on guided filtering, and they are combined into a dehazed image using a Laplacian pyramid fusion strategy based on the local connection of adaptive processing of weights. The image is then dehazed before receiving contrast and saturation improvements. The authors suggested a novel haze removal method that combines the use of the anticipated hybrid DCP module, the anticipated colour analysis (CA) module [17], and the anticipated visibility recovery (VR) module to prevent the formation of significant artifacts. Section III goes into further detail about these modules. When the collected road image has localized color-shifter light source issues, the suggested technique can effectively block out those sources of light and prevent the formation of colour shifts. The suggested procedure can more successfully eliminate haze from individual photographs taken in practical situations than existing state-of-the-art systems, according to subsequent quantitative, experimental, and qualitative evaluations.

The multi-resolution wavelet pyramid is built using the raising wavelet multiple determination technique. The issue of targets being out of alignment was resolved by an improved L-K algorithm [18]. Furthermore, the speeded up robust features (SURF) feature viewpoint fitting technique was combined at the same time. By using a multi-resolution oriented wavelet pyramid optical flow technique to decrease the likelihood of the exterior point based on the detection of feature points, difficulties with high speed, object deformation, haze, fog, uneven illumination, and limited occlusion circumstances were resolved. A real-instantaneous changing target identification system for the detection of moving targets against static backgrounds using edge detection and inter-frame difference [19]. The enhanced algorithm fixes the three-edge degree of difference edge deletion and empty phenomenon issues. The lack of the conventional three-frame differential approach is highlighted. The enhanced three-frame differential algorithm detects moving targets with more comprehensive information when merged with the Canny edge-based detection algorithm. This novel algorithm effectively makes use of the three-edge-difference methods and background elimination method for strong performances.

A brand-new automatic segmentation technique for video sequences is given that can extract moving objects [20]. The object tracker at the heart of this approach uses the Hausdorff distance to compare successive frames with the 2-D binary description of the item. The best match discovered reflects the amount of transformation the item has undertaken, and the pattern is revised in every framework to account for the replacement and formation of required changes. The preliminary model is generated repeatedly, and an innovative classical updated technique based on the idea of rearranging associated components permits quite substantial form modifications. The proposed approach is enhanced by a stationary background-removing filtering method. The analysis is the most common moving object compression strategy that has been presented recently, together with the trend of moving object compression [21]. The concepts and execution procedures of traditional moving object systems for compressions are first summarized in this study. The definitions of moving substances and their paths are then addressed concerning this. In third place, the endorsement measures for assessing the effectiveness and performance of compression processes are presented. Additionally, a few application scenarios are summarized to highlight future potential applications.

Objects can be effectively grouped into multiple classes using clustering based on the center and formerly undiscovered methods present in the dataset [22]. As location-based placement technology advances, an enhancing number of moving points are tracked, and their paths are recorded. As a result, the learning affecting object data mining will surely center on moving object trajectory clustering. The data collection method for extremely erratic raw IoT-oriented sensor data proposed in this paper uses device-to-device communication [23]. When there are significant uncertainties at the fog server, the approach iteratively locates the low-rank calculation of the dominating subspace after initially reconstructing the subspace using sample data. Moreover, the real sensor data background is estimated from the substantially erratic raw IoT-oriented sensor data stream of the traffic matrix using the resilient dominating subspace. The implementation was done for an object-tracking system with reconfigurable hardware [24] employing a productive parallel architecture. A BS-based approach is used in our implementation. To attain high system speed, the developed object tracker takes advantage of hardware parallelism. To improve our system's performance under challenging tracking situations, we additionally suggest a dual object region search strategy. The implemented hardware system used EP3SL340H1152C2 Altera Stratix III FPGA device. The software application operating on a 2.2 GHz processor is contrasted with the suggested FPGA-based implementation. For complicated visual inputs, the observed speedup can be as high as 100X.

The Kalman filter [25] is the greatest popular expectation system because it is the simplest, most effective, and easiest to use for linear measurements. To meet design criteria for embedded applications, these kinds of filter procedures are, however, tailored to hardware platforms like FPGA and GPUs. Motion detection and object tracking are addressed in this work using the multi-dimensional Kalman filter (MDKF) technique. Compared to state-of-the-art trailing algorithms trained on standard targets, the suggested tracking algorithm's numerical analysis yields competitive tracking. FPGA has been used for an effective object-detecting algorithm and FPGA implementation for real-time video [26]. The system uses the fast retina key point (FREAK) approach to characterize the key points after detecting them using the SURF procedure on individual video frames. High object detection accuracy is ensured by doing a one-to-one feature corresponding between the signifiers of the library's items and the descriptors of the video frames. Our tests show that our FPGA-based system works flawlessly and can process video frames with an 800600 resolution at 60 frames per second. Algorithms developed using common benchmarks run 23 times quicker on the suggested FPGA configuration than they do on an Intel Core i5-3210M CPU. Additionally, the ZynqTM-7000 System-on-Chip (SoC) from Xilinx is used to implement the MDKF. A multi-class classifier for binary feature vectors was created by condensing the Naive Bayes classifier [27]. It operates swiftly and effectively during both the training and testing phases because it was constructed on an FPGA with relatively few hardware resources. It was first put to the test on a dataset of handwritten digital numbers before being used in the object detection task on a specific FPGA-oriented visual surveillance system.

The object classification phases of an object recognition system are implemented by an image classifier utilizing an FPGA and random-access memory (RAM)-based distributed architecture [28]. Compared to current programmable DSP-based systems, the technology delivers a considerable performance boost. The study demonstrates how the presence of high I/O resources and pipelined architecture contributes to the significant performance gain achieved with the FPGA solution. It also serves as an example of how an FPGA solution can be used for activities with high data flow and intricate algorithmic requirements, such as real-time video processing. The RC1000-PP Virtex FPGA-based was implemented based on handle-C language. An innovative FPGA-based method for effective target recognition in hyperspectral pictures was created by the authors [29]. The Reed-Xiaoli (RX) and constrained energy minimization (CEM) algorithms are optimized using the suggested approach for streaming background statistics (SBS) methodology. The methods are popular methods for anomaly and target identification, respectively [30]. These two techniques are specifically implemented on FPGAs in a spilling mode. Most crucially, we offer a double approach that offers an adaptable datapath to choose in real-time between these two techniques, enabling the hardware to dynamically adapt to target detection or anomaly detection circumstances.

The related work presented that the work has been done in the direction of the object images and video tracking performed using MATLAB and Python simulation environments with different sizes of images, filtering, and image processing techniques [31]. Very few works have been reported in which FPGA hardware has been used for high-performance object tracking algorithms [32] and estimating the performance of the algorithm with hardware design and switching point of view. The research work presents the algorithm in that direction with the simulation environment in Xilinx ISE 14.7.

3. PROPOSED ALGORITHM

There are numerous use cases for object tracking that use various types of input footage. The techniques used to form object tracking applications are affected by whether the estimated input will be a real-time video as opposed to a prerecorded video, an image, or both. The generic method for object detection and tracking is given in Figure 1.

– Camera for images: real-time images/video streams from practically any camera can be used to apply modern object-tracking techniques. Consequently, object tracking can be done using the video stream from a USB camera or an IP camera bypassing the individual frames to a tracking algorithm. With real-time video inputs from one or more cameras, frame skipping, or parallelized processing are frequent techniques to enhance object tracking performance [33].

- Image/video pre-processing: image/video consists of a sequence of frames. The individual frame depicts a different state of an object's status. Object detection at the beginning of the frame and continued tracking of that specific object throughout the video sequence [34].
- Object detection: the term "object detection" refers to a category of computer technology that searches movies and digital images for occurrences of semantic objects belonging to a particular class such as automobiles, buildings, and people. BS, optical flow, and frame differencing are some basic methods for object detection.
- Object localization: object localization is the process of identifying the type of object of interest in each detected object in the frame. It is necessary to determine what kind of object it is. Several elements, such as texture, colour, motion, and shape, can be used to identify an object. It might be shape-based, texture-based, color-based, or motion-based, depending on the variables put into play
- Object tracking: using successive image frames to monitor an object, object tracking is a technique for figuring out how an object moves concerning other things. The most common technique is to gauge how much the object's centroid has moved in (X, Y) between frames. The three methods of object tracking are point-oriented tracking, kernel-based tracking, and silhouette-based tracking [35].

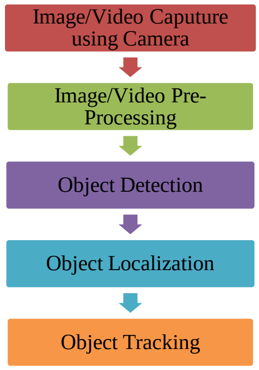


Figure 1. Object detection and tracking

A computer vision and image processing feature called the histogram of oriented gradients (HoG) is applied to identify objects. Using a detection window, or region of interest (RoI), the HoG descriptor method considers instances of gradient induction in focused areas of an image. The behaviour of the HoG is shown in Figure 2 in which the entire image is processed using a (4×4) mask. The HoG signifier algorithm implementation strategy [36] and methodology are shown in Figure 3.

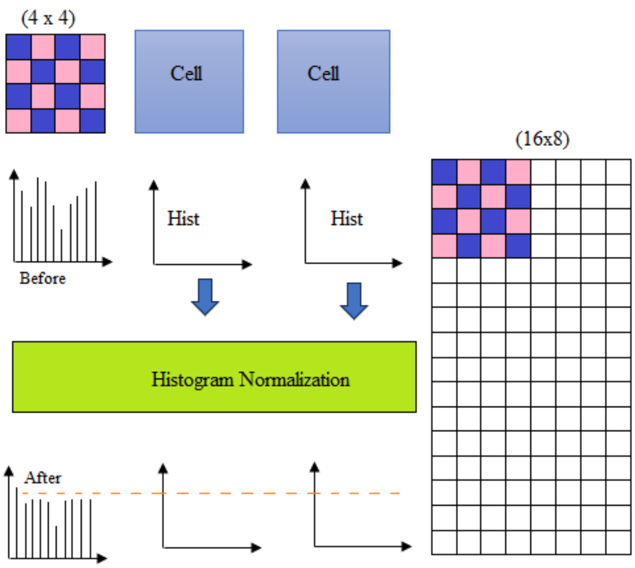


Figure 2. HoG execution of the images

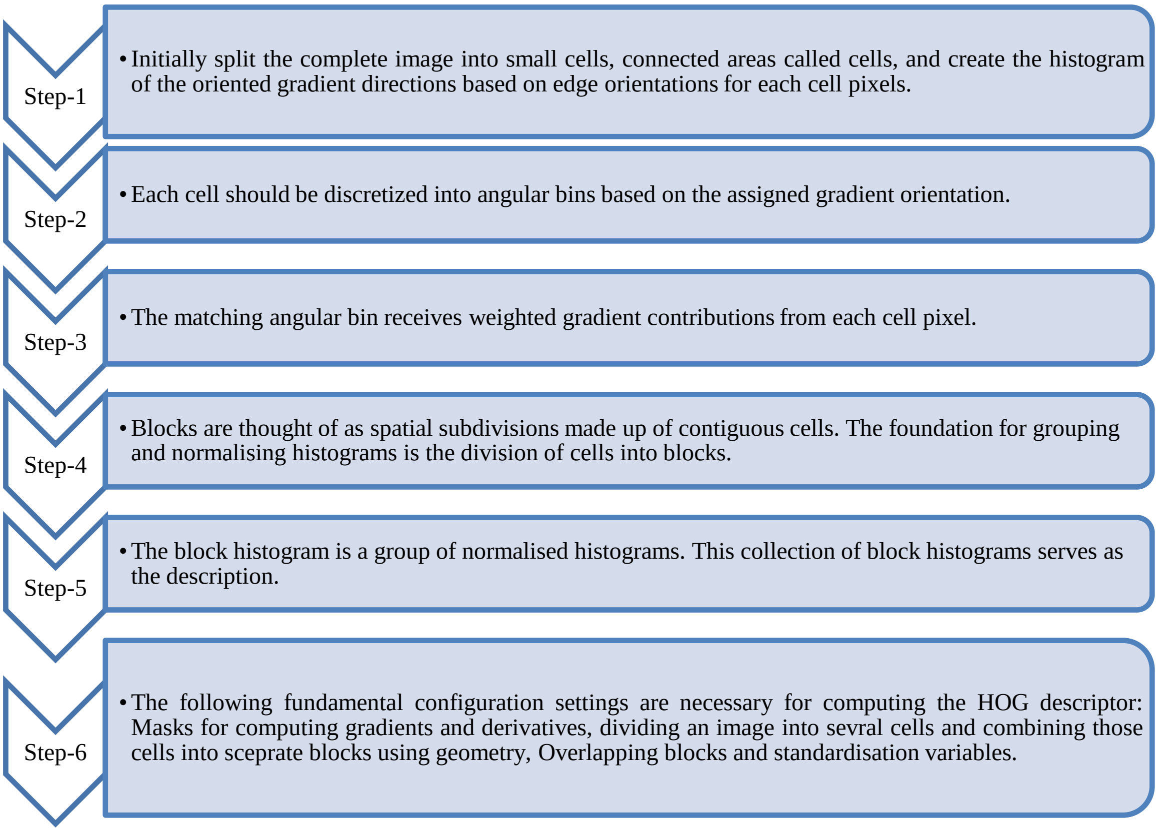


Figure 3. Methodology

Consider the input image from which HoG characteristics must be calculated. Resize the image to 128×64 pixels (128 pixels in height and 64 pixels in width). This dimension may apply to the type of detection required to provide better results in terms of object detection. It is necessary to compute the gradient of the pictures, which is obtained by combining the magnitude and angle from the image. For each pixel in a (4×4) block, G_x and G_y are first calculated. First, G_x and G_y are determined for each pixel value using the mathematical (1) and (2).

$$G_x\,(r_d, C_d) = I_m\,(r_d, C_d + 1) - I_m\,(r_d, C_d - 1) \tag{1}$$

$$G_y\,(r_d, C_d) = I_m\,(r_d - 1, C_d) - I_m\,(r_d + 1, C_d) \tag{2}$$

Here r_d and C_d present the row and column data processing. After estimation, the values of G_x and G_y , the values of magnitude and phase are calculated using (3) and (4) respectively.

$$Magnitude, M = \sqrt{G_x^2 + G_y^2} \tag{3}$$

$$Phase\ , \emptyset = \left|\tan^{-1}\left(\frac{G_y}{G_x}\right)\right| \tag{4}$$

After collecting the gradient of individual pixels, the gradient matrices with magnitude value and angle values matrix are grouped into 4×4 cells to make a block, and boundaries and centers are decided to estimate the feature vectors.

4. RESULTS AND DISCUSSIONS

The chip view of the HoG algorithm is shown in Figure 4. The description of all the pins utilized is given to understand the input and output signals of the design. Register transfer level (RTL) provides a relatively low degree of abstraction, which makes it possible to describe digital circuits without much difficulty. The RTL consists of the clk signal and reset signal as the main inputs. The RTL is extracted from the Xilinx ISE simulation tool.

Image_in_Histogram_Pixel_0<31:0> presents the pixel0 intensity input integer before processing in histogram equalization. Image_in_Histogram_Pixel_1 <31:0> presents the pixel1 intensity input integer before processing in histogram equalization. Image_in_Histogram_Pixel_2 <31:0> presents the pixel2 intensity input integer before processing in histogram equalization. Image_in_Histogram_Pixel_3 <31:0> presents the pixel3 intensity input integer before processing in histogram equalization. Image_in_Histogram_Pixel_4 <31:0> presents the pixel4 intensity input integer before processing in histogram equalization. Image_in_Histogram_Pixel_5 <31:0> presents the pixel5 intensity input integer before processing in histogram equalization. Image_in_Histogram_Pixel_6 <31:0> presents the pixel6 intensity input integer before processing in histogram equalization. Image_in_Histogram_Pixel_7 <31:0> presents the pixel7 intensity input integer before processing in histogram equalization. Image_in_Histogram_Pixel_8 <31:0> presents the pixel8 intensity input integer before processing in histogram equalization. Image_in_Histogram_Pixel_9 <31:0> presents the pixel9 intensity input integer before processing in histogram equalization.

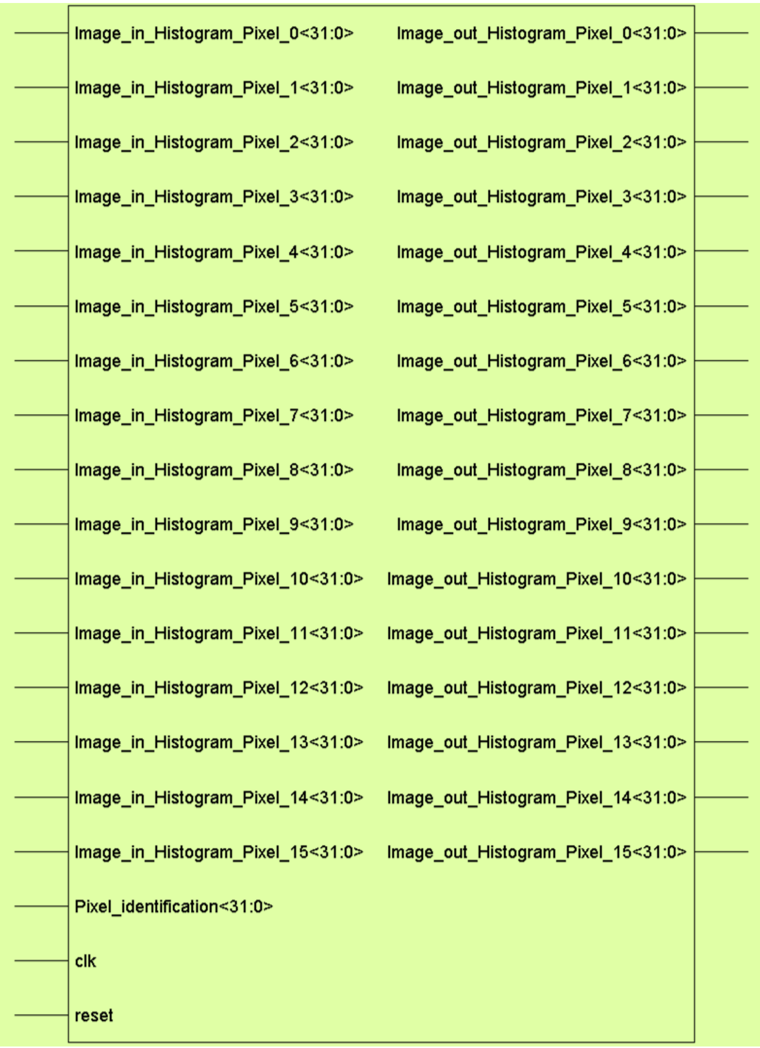


Figure 4. RTL of the HoG chip design

Image_in_Histogram_Pixel_10 <31:0> presents the pixel10 intensity input integer before processing in histogram equalization. Image_in_Histogram_Pixel_11 <31:0> presents the pixel11 intensity input integer before processing in histogram equalization. Image_in_Histogram_Pixel_12 <31:0> presents the pixel12 intensity input integer before processing in histogram equalization. Image_in_Histogram_Pixel_13 <31:0>

presents the pixel13 intensity input integer before processing in histogram equalization. Image_in_Histogram_Pixel_14 <31:0> presents the pixel14 intensity input integer before processing in histogram equalization. Image_in_Histogram_Pixel_15 <31:0> presents the pixel15 intensity input integer before processing in histogram equalization.

Image_out_Histogram_Pixel_0 <31:0> presents the pixel0 intensity output integer after processing in histogram equalization. Image_out_Histogram_Pixel_1 <31:0> presents the pixel1 intensity output integer after processing in histogram equalization. Image_out_Histogram_Pixel_2 <31:0> presents the pixel2 intensity output integer after processing in histogram equalization. Image_out_Histogram_Pixel_3 <31:0> presents the pixel3 intensity output integer after processing in histogram equalization. Image_out_Histogram_Pixel_4 <31:0> presents the pixel4 intensity output integer after processing in histogram equalization. Image_out_Histogram_Pixel_5 <31:0> presents the pixel5 intensity output integer after processing in histogram equalization. Image_out_Histogram_Pixel_6 <31:0> presents the pixel6 intensity output integer after processing in histogram equalization. Image_out_Histogram_Pixel_7 <31:0> presents the pixel7 intensity output integer after processing in histogram equalization. Image_out_Histogram_Pixel_8 <31:0> presents the pixel8 intensity output integer after processing in histogram equalization. Image_out_Histogram_Pixel_9 <31:0> presents the pixel9 intensity output integer after processing in histogram equalization. Image_out_Histogram_Pixel_10 <31:0> presents the pixel10 intensity output integer after processing in histogram equalization. Image_out_Histogram_Pixel_11 <31:0> presents the pixel11 intensity output integer after processing in histogram equalization. Image_out_Histogram_Pixel_12 <31:0> presents the pixel12 intensity output integer after processing in histogram equalization. Image_out_Histogram_Pixel_13 <31:0> presents the pixel13 intensity output integer after processing in histogram equalization. Image_out_Histogram_Pixel_14 <31:0> presents the pixel14 intensity output integer after processing in histogram equalization. Image_out_Histogram_Pixel_15 <31:0> presents the pixel15 intensity output integer after processing in histogram equalization. Clock is the input given to assign the positive edge of the clock signal and reset will provide the reset of all the pixel values. Figure 5 presents the Modelsim simulation of HoG for object tracking for test-1 and test-2 in integer real value of pixels. Figure 6 shows the Modelsim simulation of HoG for object tracking for test-1 and test-2 in binary value of pixels. Table 1 presents the lists of test-1 (Image_in_Histogram_Pixel_ and Image_out_Histogram_Pixel_) and Table 2 presents the lists of test-2 (Image_in_Histogram_Pixel_ and Image_out_Histogram_Pixel_).

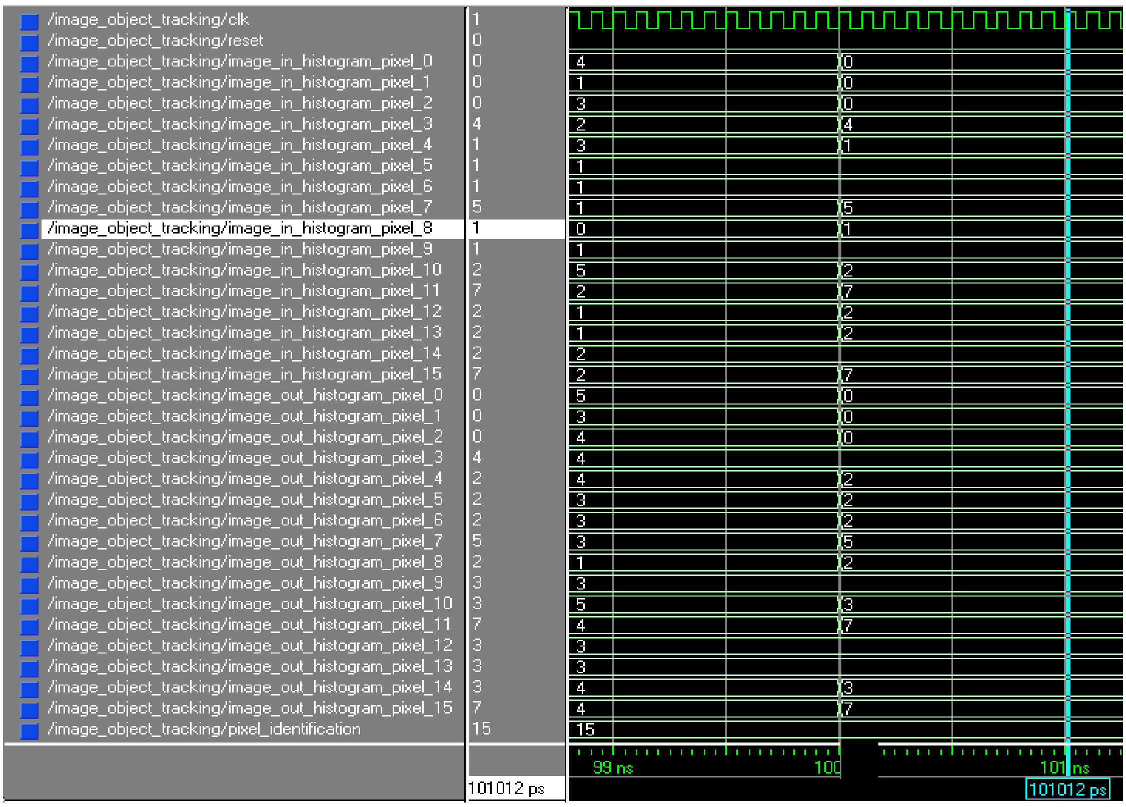


Figure 5. Modelsim simulation of HoG for object tracking for test-1 and test-2 in integer real value of pixels

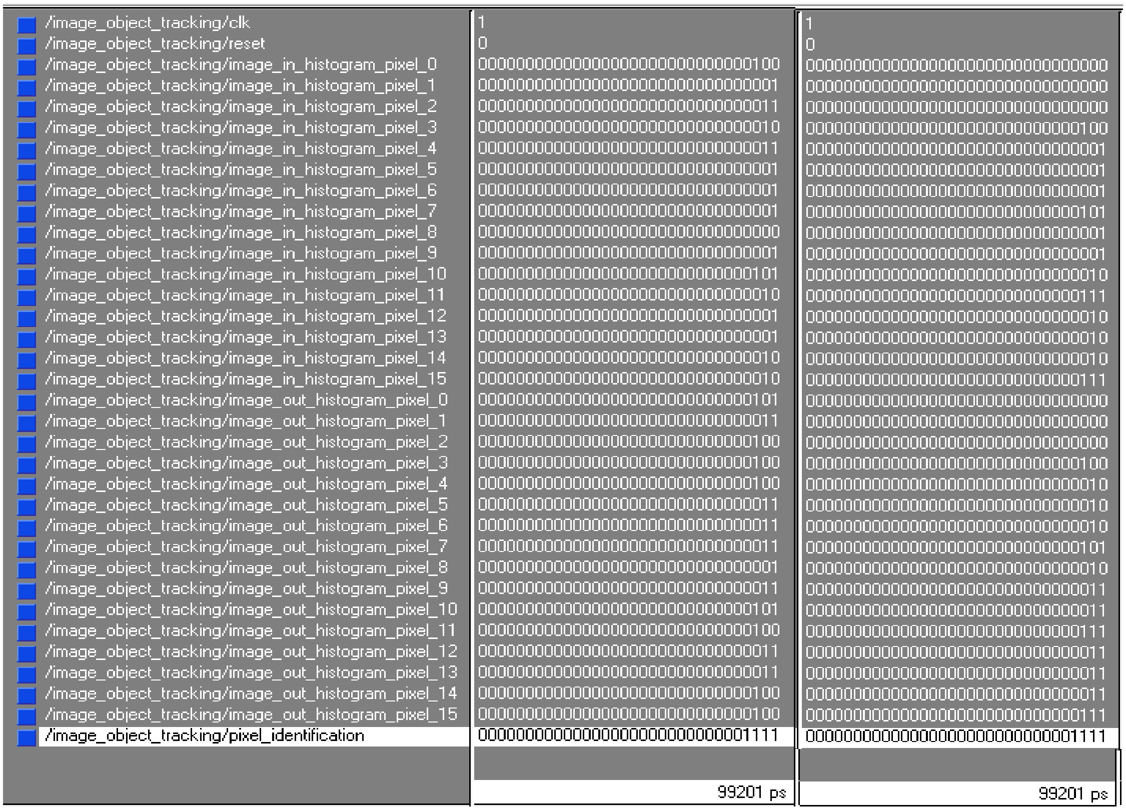


Figure 6. Modelsim simulation of HoG for object tracking for test-1 and test-2 in binary value of pixels

Table 1. Test-1

Image_in_Histogram_Pixel_	Image_out_Histogram_Pixel_
0=4	0=5
1=1	1=3
2=3	2=4
3=2	3=4
4=3	4=4
5=1	5=3
6=1	6=3
7=1	7=3
8=0	8=1
9=1	9=3
10=5	10=5
11=2	11=4
12=1	12=3
12=1	12=3
14=2	14=4
15=2	15=4

Table 2. Test-2

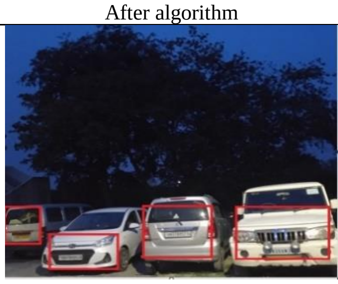
Image_in_Histogram_Pixel_	Image_out_Histogram_Pixel_
0=0	0=0
1=0	1=0
2=0	2=0
3=4	3=4
4=1	4=2
5=1	5=2
6=1	6=2
7=5	7=5
8=1	8=2
9=1	9=3
10=2	10=3
11=7	11=7
12=2	12=3
12=2	12=3
14=2	14=3
15=7	15=7

The response time of the image is analyzed in MATLAB and Xilinx ISE 14.7. Table 3 lists the description of the response time for these simulations. Table 4 presents the simulation outcome of the algorithm applied for the random images/videos taken from the author’s camera.

Table 3. Comparison of the response time for detection

Description	Response time in MATLAB (seconds)	Response time in Xilinx ISE in (nanoseconds)
Object image/Video-1	0.39	0.672
Object image/Video-2	0.45	0.428
Object image/Video-3	0.42	0.512
Object image/Video-4	0.67	0.905
Object image/Video-5	0.72	1.005

Table 4. Simulation of the sampled image/video

S. No	Original	After algorithm
Image/Video-1		
Image/Video-2		
Image/Video-3		
Image/Video-4		
Image/Video-5		

5. CONCLUSIONS

Real-time FPGA-based object tracking is frequently employed in a variety of applications, including video surveillance, human-computer interaction, traffic monitoring, and vehicle navigation. Various algorithms based on feature descriptors, optical flow, template matching, or texture operators are used. Most often, the algorithms on the FPGA track every moving object or just certain types of objects. The simulation of the HoG hardware chip is done successfully in Xilinx ISE 14.7 software which is used to identify objects. The behavior of the chip simulations is verified using Modelsim 10.0 for object detection. The maximum response time estimation on FPGA is 1.005 ns which is much less in comparison to MATLAB response time of 0.72 seconds. The maximum frequency support in the design is reported as 315 MHz. The comparative performance for the chip estimates that FPGA targeted simulation in Xilinx provided optimal delay in comparison to MATLAB response time.

REFERENCES

[1] C. Blair, N. M. Robertson, and D. Hume, "Characterizing a heterogeneous system for person detection in video using histograms of oriented gradients: power versus speed versus accuracy," *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, vol. 3, no. 2, pp. 236–247, Jun. 2013, doi: 10.1109/JETCAS.2013.2256821.

[2] J. G. Pandey, A. Karmakar, C. Shekhar, and S. Gurunarayanan, "An FPGA-based architecture for local similarity measure for image/video processing applications," in *2015 28th International Conference on VLSI Design*, IEEE, Jan. 2015, pp. 339–344, doi: 10.1109/VLSID.2015.63.

[3] H.-Y. Cheng and J.-N. Hwang, "Integrated video object tracking with applications in trajectory-based event detection," *Journal of Visual Communication and Image Representation*, vol. 22, no. 7, pp. 673–685, Oct. 2011, doi: 10.1016/j.jvcir.2011.07.001.

[4] A. Cavallaro, O. Steiger, and T. Ebrahimi, "Tracking video objects in cluttered background," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 15, no. 4, pp. 575–584, Apr. 2005, doi: 10.1109/TCSVT.2005.844447.

[5] H. Zhu, H. Wei, B. Li, X. Yuan, and N. Kehtarnavaz, "A review of video object detection: datasets, metrics and methods," *Applied Sciences*, vol. 10, no. 21, p. 7834, Nov. 2020, doi: 10.3390/app10217834.

[6] R. S. Adesh Kumar, Pankil Ahuja, "Text extraction and recognition from an image using image processing in MATLAB," *Atlantis Press*, vol. 2013, no. Cac2s, pp. 429–435, 2013.

[7] A. Goel, A. K. Goel, and A. Kumar, "The role of artificial neural network and machine learning in utilizing spatial information," *Spatial Information Research*, vol. 31, no. 3, pp. 275–285, Jun. 2023, doi: 10.1007/s41324-022-00494-x.

[8] A. Goel, A. K. Goel, and A. Kumar, "Performance analysis of multiple input single layer neural network hardware chip," *Multimedia Tools and Applications*, vol. 82, no. 18, pp. 28213–28234, Jul. 2023, doi: 10.1007/s11042-023-14627-3.

[9] J. Li, K.-F. Un, W.-H. Yu, P.-I. Mak, and R. P. Martins, "An FPGA-based energy-efficient reconfigurable convolutional neural network accelerator for object recognition applications," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 68, no. 9, pp. 3143–3147, Sep. 2021, doi: 10.1109/TCSII.2021.3095283.

[10] C. B. Murthy, M. F. Hashmi, N. D. Bokde, and Z. W. Geem, "Investigations of object detection in images/videos using various deep learning techniques and embedded platforms—a comprehensive review," *Applied Sciences*, vol. 10, no. 9, p. 3280, May 2020, doi: 10.3390/app10093280.

[11] P. Thiruvikraman, T. A. Kumar, R. Rajmohan, and M. Pavithra, "A survey on haze removal techniques in satellite images," *Irish Interdisciplinary Journal of Science and Research*, vol. 5, no. 2, pp. 1–6, 2021.

[12] J. Xie, E. Stensrud, and T. Skramstad, "Detection-based object tracking applied to remote ship inspection," *Sensors*, vol. 21, no. 3, p. 761, Jan. 2021, doi: 10.3390/s21030761.

[13] S. B., A. J. Tom, and S. N. George, "Simultaneous denoising and moving object detection using low rank approximation," *Future Generation Computer Systems*, vol. 90, pp. 198–210, Jan. 2019, doi: 10.1016/j.future.2018.07.065.

[14] A. S. Rawat, A. Rana, A. Kumar, and A. Bagwari, "Application of multi layer artificial neural network in the diagnosis system: a systematic review," *IAES International Journal of Artificial Intelligence (IJ-AI)*, vol. 7, no. 3, pp. 138-142, Aug. 2018, doi: 10.11591/ijai.v7.i3.pp138-142.

[15] M. Sarkar, P. Sarkar Rakshit, U. Mondal, and D. Nandi, "Tetrolet transform and dual dictionary learning-based single image fog removal," *Arabian Journal for Science and Engineering*, vol. 48, no. 8, pp. 10771–10786, Aug. 2023, doi: 10.1007/s13369-023-07681-4.

[16] S. Ma *et al.*, "Image dehazing based on improved color channel transfer and multiexposure fusion," *Advances in Multimedia*, vol. 2023, pp. 1–10, May 2023, doi: 10.1155/2023/8891239.

[17] S.-C. Huang, B.-H. Chen, and Y.-J. Cheng, "An efficient visibility enhancement algorithm for road scenes captured by intelligent transportation systems," *IEEE Transactions on Intelligent Transportation Systems*, vol. 15, no. 5, pp. 2321–2332, Oct. 2014, doi: 10.1109/TITS.2014.2314696.

[18] L. Dan, J. Dai-Hong, B. Rong, S. Jin-Ping, Z. Wen-Jing, and W. Chao, "Moving object tracking method based on improved lucas-kanade sparse optical flow algorithm," in *2017 International Smart Cities Conference (ISC2)*, IEEE, Sep. 2017, pp. 1–5, doi: 10.1109/ISC2.2017.8090850.

[19] L. Gang, N. Shangkun, Y. Yugan, W. Guanglei, and Z. Siguo, "An improved moving objects detection algorithm," in *2013 International Conference on Wavelet Analysis and Pattern Recognition*, IEEE, Jul. 2013, pp. 96–102, doi: 10.1109/ICWAPR.2013.6599299.

[20] T. Meier and K. N. Ngan, "Automatic segmentation of moving objects for video object plane generation," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 8, no. 5, pp. 525–538, 1998, doi: 10.1109/76.718500.

[21] P. Sun, S. Xia, G. Yuan, and D. Li, "An overview of moving object trajectory compression algorithms," *Mathematical Problems in Engineering*, vol. 2016, pp. 1–13, 2016, doi: 10.1155/2016/6587309.

[22] G. Yuan, P. Sun, J. Zhao, D. Li, and C. Wang, "A review of moving object trajectory clustering algorithms," *Artificial Intelligence Review*, vol. 47, no. 1, pp. 123–144, Jan. 2017, doi: 10.1007/s10462-016-9477-7.

[23] S. Sanyal and P. Zhang, "Improving quality of data: IoT data aggregation using device to device communications," *IEEE Access*, vol. 6, pp. 67830–67840, 2018, doi: 10.1109/ACCESS.2018.2878640.

[24] S. Liu, A. Papakonstantinou, H. Wang, and D. Chen, "Real-time object tracking system on FPGAs," in *2011 Symposium on Application Accelerators in High-Performance Computing*, IEEE, Jul. 2011, pp. 1–7, doi: 10.1109/SAAHPC.2011.22.

[25] P. Babu and E. Parthasarathy, "FPGA implementation of multi-dimensional Kalman filter for object tracking and motion detection," *Engineering Science and Technology, an International Journal*, vol. 33, 2022, doi: 10.1016/j.jestch.2021.101084.

[26] J. Zhao, X. Huang, and Y. Massoud, "An efficient real-time FPGA implementation for object detection," in *2014 IEEE 12th International New Circuits and Systems Conference (NEWCAS)*, 2014, pp. 313–316, doi: 10.1109/NEWCAS.2014.6934045.

[27] H. Meng, K. Appiah, A. Hunter, and P. Dickinson, "FPGA implementation of Naive Bayes classifier for visual object recognition," in *CVPR 2011 WORKSHOPS*, IEEE, Jun. 2011, pp. 123–128, doi: 10.1109/CVPRW.2011.5981831.

[28] P. McCurry, F. Morgan, and L. Kilmartin, "Xilinx FPGA implementation of an image classifier for object detection applications," in *Proceedings 2001 International Conference on Image Processing (Cat. No.01CH37205)*, IEEE, pp. 346–349, doi: 10.1109/ICIP.2001.958122.

[29] B. Yang, M. Yang, A. Plaza, L. Gao, and B. Zhang, "Dual-mode FPGA implementation of target and anomaly detection algorithms for real-time hyperspectral imaging," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 8, no. 6, pp. 2950–2961, Jun. 2015, doi: 10.1109/JSTARS.2015.2388797.

[30] Y. Dong, L. Liu, J. Xu, and G. Wan, "Target detection algorithm based on improved homomorphic filter in haze days," in *2022 Global Reliability and Prognostics and Health Management (PHM-Yantai)* Oct. 2022, pp. 1–5, doi: 10.1109/PHM-Yantai55411.2022.9942118.

[31] A. Kumar, P. Rastogi, and P. Srivastava, "Design and FPGA implementation of DWT, image text extraction technique," *Procedia Computer Science*, vol. 57, pp. 1015–1025, 2015, doi: 10.1016/j.procs.2015.07.512.
[32] A. Kumar, "Study and analysis of different segmentation methods for brain tumor MRI application," *Multimedia Tools and Applications*, vol. 82, no. 5, pp. 7117–7139, Feb. 2023, doi: 10.1007/s11042-022-13636-y.
[33] T. Barbu, "Pedestrian detection and tracking using temporal differencing and HOG features," *Computers & Electrical Engineering*, vol. 40, no. 4, pp. 1072–1079, May 2014, doi: 10.1016/j.compeleceng.2013.12.004.
[34] C.-W. Liang and C.-F. Juang, "Moving object classification using local shape and HOG features in wavelet-transformed space with hierarchical SVM classifiers," *Applied Soft Computing*, vol. 28, pp. 483–497, Mar. 2015, doi: 10.1016/j.asoc.2014.09.051.
[35] G. Jemilda and S. Baulkani, "Moving object detection and tracking using genetic algorithm enabled extreme learning machine," *International Journal of Computers Communications & Control*, vol. 13, no. 2, pp. 162–174, Apr. 2018, doi: 10.15837/ijccc.2018.2.3064.
[36] A. Devrari and A. Kumar, "Reconfigurable linear feedback shift register for wireless communication and coding," *International Journal of Reconfigurable and Embedded Systems (IJRES)*, vol. 12, no. 2, pp. 195-204, Jul. 2023, doi: 10.11591/ijres.v12.i2.pp195-204.

BIOGRAPHIES OF AUTHORS

Monika Sharma is currently working as a research scholar in computer science and engineering at Galgotia's University, Noida, NCR, India. She is M.Tech. in computer science and engineering and B.Tech. in computer science and engineering in 2012 and 2009 respectively. She has published more than 20 research papers and book chapters. She is working as a lecturer at Government Girls Polytechnic College, Daurala, Meerut, under the Uttar Pradesh Technical Department, India. She is having experience of 14 years. She can be contacted at email: monikasharma.cse@gmail.com.

Prof. (Dr.) Kuldeep Singh Kaswan is presently working in the School of Computing Science and Engineering, Galgotia's University, Uttar Pradesh, India. His contributions focus on BCI, cyborg, and data science. His Academic degrees and thirteen years of experience working with global Universities like, Amity University, Noida, Gautam Buddha University, Greater Noida, and PDM University, Bahadurgarh, have made him more receptive and prominent in his domain. He received a doctorate in computer science from Banasthali Vidyapith, Rajasthan. He received a doctor of engineering (D.Engg.) from Dana Brain Health Institute, Iran. He has obtained Master degree in computer science and engineering from Choudhary Devi Lal University, Sirsa (Haryana). He has supervised many UG and PG projects of engineering students. He has supervised 3 Ph.D. graduates and presently supervising 4 Ph.D. students. He is also a member of IEEE, Computer Science Teacher Association (CSTA), New York, USA, International Association of Engineers (IAENG), Hong Kong, International Association of Computer Science and Information Technology (IACSIT), USA, professional member Association of Computing Machinery, USA. He has published 9 Books and 40 book chapters at National/International level. He has the number of publications also in International/National Journal and Conferences. He is an editor/author, and review editor of Journals and Books with IEEE, Wiley, Springer, IGI, and River. He can be contacted at email: kaswankuldeep@gmail.com.

Dr. Dileep Kumar Yadav received the engineering degree (B.Tech. in computer science and engineering) from Uttar Pradesh Technical University, Lucknow, UP, India in 2006 and master's degree (M.Tech. in computer science and technology) from School of Computer and Systems Sciences, Jawaharlal Nehru University, New Delhi, India in 2011. Dr. Yadav has earned Ph.D. (computer science and technology) degree from School of Computer and Systems Sciences, Jawaharlal Nehru University New Delhi, India in 2016. He is Sun Certified Java Programmer. He is author of 65 research publications (Patent, Journals (SCI/SCIE/SCOPUS), and National/International conferences). He has also written books and many book chapters for internationally reputed publishers. His primary research interests are in image processing, computer vision, and blockchain security using artificial intelligence and machine learning over dynamic data. Dr. Yadav supervised various students of master's degree and Ph.D. Dr. Yadav is also associated with many international journals as associate editor, member, Int. Editorial Board member etc. He has more than 12 years of working experience in industry as well as academia. Dr. Yadav is the recipient of various awards from national and International organizations in research. He is also supervising many national and international students to pursue their research work. Currently, Dr. Yadav is working as an associate professor in the Department of CSE, SCSET, Bennett University, Greater Noida, India. He can be contacted at email: dileep25200@gmail.com.