



## Target Detection in Video Images Using HOG-Based Cascade Classifier

Ayad A. Namah<sup>1</sup>, Nabeel M. Mirza<sup>1\*</sup>, Ali A. Al-Zuky<sup>2</sup>

<sup>1</sup> Department of Physics, College of Education, Mustansiriyah University, Baghdad 10052, Iraq

<sup>2</sup> Department of Physics, College of Science, Mustansiriyah University, Baghdad 10052, Iraq

Corresponding Author Email: [nabeel.mirza@uomustansiriyah.edu.iq](mailto:nabeel.mirza@uomustansiriyah.edu.iq)

<https://doi.org/10.18280/ria.360507>

### ABSTRACT

**Received:** 1 August 2022

**Accepted:** 18 October 2022

#### Keywords:

*training, AdaBoost algorithm, Cascade classifier, Histogram of Oriented Gradients (HOG), imresize*

Detecting small objects using computer vision is a challenging task due to their small size in the image and therefore the lack of features when describing them. In this paper, a computer was trained to detect three small balls using 20 levels of the AdaBoost cascade classifier. The features of the balls in each level are described using the HOG feature descriptor. Three balls were recorded in practice at various distances ( $d = 2, 3, 4, \dots, 10$  m) from the camera and the targets (balls). The frames are then taken from the videos and resized using five magnification factors ( $RS = 1, 3, 5, 7, \text{ and } 9$ ) to make the balls seem as they should. According to the results, the detection rate of balls at all distances was 80% when using the magnification factor  $RS = 1$ , 90% when using the magnification factor  $RS = 3, 5, \text{ and } 7$ , and 100% when using the magnification factor  $RS = 9$ . The suggested approach was also used in calculating the height and width of the detected balls. The overall results indicated that the height and width of the balls dwindle as the distance between the camera and the targets increases.

## 1. INTRODUCTION

Computer vision is a branch of research that enables machines to process, analyze, and interpret the content of digital images and videos in order to perceive visuals similarly to humans. It lets a computer recognize and detect objects in images, in other words [1, 2]. Object detection is determining a possible location of region of interest (ROI) in input image or in real time video [3, 4]. It is basic step to all analysis methods such as object recognition and object tracking. The objective of object detection is to determine the location of ROI in digital images and surround it with a bounding box [5]. Object detection is the most challenging task in computer vision as it is deal with detecting instances of visual objects in digital images, determine object angle, with respect to lighting condition of the scene contained the ROI. Hence, a computational models and techniques developed with the help of computer vision to enable computer understanding digital images and react suitably [6]. Numerous machine learning-based computer vision algorithms have been developed to detect objects in image such as cascade classifier to detect objects. A cascade of classifiers is a degenerated decision tree consist of levels of increasing complexity where at first level a classifier is trained to detect almost all ROI and triggers evaluation of classifier of second level which is also been modified to achieve very high detection rate. A positive result from the second level triggers a third level, and so on [7]. Object detection is one of the vital research topics as it used wide range of application such as intelligent surveillance [8, 9], home automation [10], manufacturing [11], and healthcare [12]. There is other common application like face detection [13], character recognition [14], autonomous driving [15], hand gesture recognition [16], etc.

As the widespread range of object detection application,

many researchers developing different methods and algorithm in this topic such as: Joseph and Pradeep [17] suggested a method to detect and track object in a video. The detection of the object performed using supported vector machine SVM classifier. Where, the Histogram of Oriented Gradients (HOG) feature is used beside with the classifier to distinguish object from the background. While Shah et al. [18] proposed an algorithm to detect green ball in real-time video using image processing techniques. They employed Gaussian blur to identify the color of the ball, and a contour of the identified color of the object to detect its shape. However, in scenarios with busy backgrounds, a false ball detection occurred. Chowdhury et al. [19] suggested an algorithm to detect cups in images based on Viola Jones algorithm. During the training stage, they applied their algorithm to a collection of positive and negative datasets that they downloaded from the internet. The detection accuracy of their suggested approach was improved by increasing the positive and negative datasets. Sultana et al. [20] addressed a method to detect object in image using HOG feature and template matching technique. Where, the cross-correlation between HOG feature descriptor and the template is used. However, they note that the results are poor when the template image objects are not part of the original image. Nugraha et al. [21] introduced a ball detection system using humanoid robot. The camera in robot captured frame of the ball, then the ball detected basing on a computer-training model. Their detection model trained using YOLOv3 (you look only once) method. The ball could be detected by their suggested technique up to 90 cm away from the robot.

This work aims to track and detect objects at various ranges, such balls or spherical forms. These targets may be seen in the images and videos that were taken. It also focuses on analyzing and estimating the ball's height and width in relation to its distance from the observation position.

## 2. TARGET DETECTION

Ball is considered as small object and can be detect using cascade learning method. The computer is trained to detect ball using cascade of multiple levels. In each level, ball features extracted using histogram of oriented gradients (HOG) feature descriptor considered as weak features. Weak features are weak classifiers obtained by weak learners that the AdaBoost algorithm can produce. Here, each weak learner is used to select a single HOG feature (the best feature to separate positive and negative samples). To select a small number of features, AdaBoost is used, which is based on a cascade of weak classifier level by level. All training samples, whether positive and negative, start off with the same particular value termed "weight." The weak classifier is trained using a single feature in the initial iteration of the algorithm. The weights are changed for the second iteration (the new selected feature), and the samples that the first iteration incorrectly categorised receive greater weights [7, 22, 23]. AdaBoost algorithm is used along with cascade classifier to combine the weak features that extracted from each level in order to obtain strong feature used to classify ball in input images [19].

### 2.1 HOG feature descriptor

The HOG is a feature descriptor introduced by Dalal and Triggs [24]. It is used to characterize ROI in an input image by pixel gradient distribution to conceive set of features [25, 26]. The essential idea of HOG is the division of an input image into blocks of size 8x8 pixels. The gradient magnitude and orientation for each block's individual pixels are constructed. A 9-bin (ranging from 0° to 180° with a step of 20°) histogram of the oriented gradient computation for each block in the image based on the orientation and magnitude matrices Each bin is a vector feature [27]. It is also helpful to contrast-normalize the local responses before utilizing them for improved invariance to light, shadowing, etc. To do this, it is possible to collect a measure of local histogram "energy" over a greater number of spatial areas (blocks), and then use the data to normalize each cell in the block. The HOG descriptors are normalized descriptor blocks in which each detection window is split into cells with a size of 8 by 8 pixels for the input image and each group of 2 x 2 cells (16 x 16) is integrated into a block in a sliding manner so that blocks overlap [24]. Each cell consists of a 9-bin HOG, and each block contains a concatenated vector of all its cells. Each block is thus represented by a 36-features vector that is normalized to an L<sup>2</sup> unit length. Each 64 × 128 detection window is represented by 7 × 15 blocks, giving a total of 3780 features per detection window [24, 28]. The HOG features for an input image illustrated in Figure 1.



Figure 1. Sample of the input image and its HOG

### 2.2 Boosted cascade classifier

Cascade classifier is a special case of ensemble learning. It is used to detect objects by extracting set of object features using multiple levels. Where, each level is an ensemble of weak classifiers [27]. The ROI detected by sliding window over the entire input image. At each level, the classifier trained to label all ROI and background in image using single feature. If the value of the feature corresponds a sub-region, then the region is classified as positive (containing ROI). Otherwise, the sub-region classified as negative (background). The positive results from current stage triggers evaluation of classifier of next stage. While, the negative results are rejected immediately and the classification of the region with negative results is completed [29]. The overall form of detection process of all levels is called "cascade". The cascade considered as single classifier as combine the weak classifier of all levels. AdaBoost algorithm [30] is used to combine weak classifiers from all levels to obtain strong classifier. The strong classifier is strong features used to detect ROI in images or real-time video [2]. The weak classifiers are obtained in a series of sequentially re-weighted updates of the training data [24]. According to [7], a weak classifier ( $F_w$ ) using a single Haar feature ( $f$ ) is as follows:

$$F_w(x) = \begin{cases} 1 & pf(x) < p_{th} \\ 0 & otherwise \end{cases} \quad (1)$$

where:  $x$  is a 24×24 sub-window, ( $th$ ) is the threshold that decides whether ( $x$ ) should be classified as a positive object (ball) or a negative object (non-ball), and  $p \in \{1,-1\}$  is a polarity term indicating the direction of inequality. Using the weighted sum of all weak classifiers, a single AdaBoost combines all weak classifiers (weak features) from all iterations to produce one strong classifier (strong feature) [24].

## 3. IMAGING SETUP AND TOOLS USED

In all, nine videos of three blue balls were recorded; the videos differ by the distance between the balls and the camera. As seen in Figure 2, the equipment utilized to record videos is as follows:

- A 13 megapixels, f/1.9 aperture Sony Exmor RS IMX258 rear camera of Samsung Galaxy J7 Prime mobile.
- Stand of height (170 Cm).
- Three similar blue balls of circumference (62 Cm).



Figure 2. Utilized tools

The imaging system setup in this work (illustrated in Figure 3) was done in a, (44m length and 24m width), five-a-side football stadium located in Diyala Governorate, Muqdadiya District. Where the camera used to recorded nine videos (vid1, vid2, vid3, ..., vid9) of three blue balls set in horizontal level

at distance 1m between each other. The videos differ by differing the distance (d) between camera and balls level. The distances (d) between camera and ball used to recorded the videos were: vid1 at d=2m, vid2 at d=3m, vid3 at d=4m, ..., vid9 at d=10m). Hence, each video is recorded at a fixed specific distance between camera and balls level. Figure 4 shows nine frames extracted from the nine videos to clarify the variable distance between the camera and the balls. The frames have the same size and viewing angle regardless of how far apart the balls are from the camera, whereas the resolution of the images varies depending on the change in the distance between the camera and the balls. The videos recorded in a day lighting on 16/11/2021 at 03:30 pm.

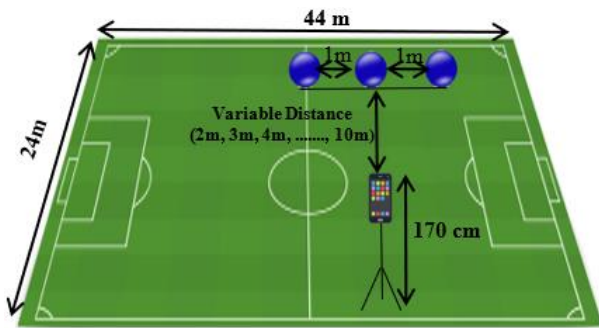


Figure 3. Diagram of imaging system setup

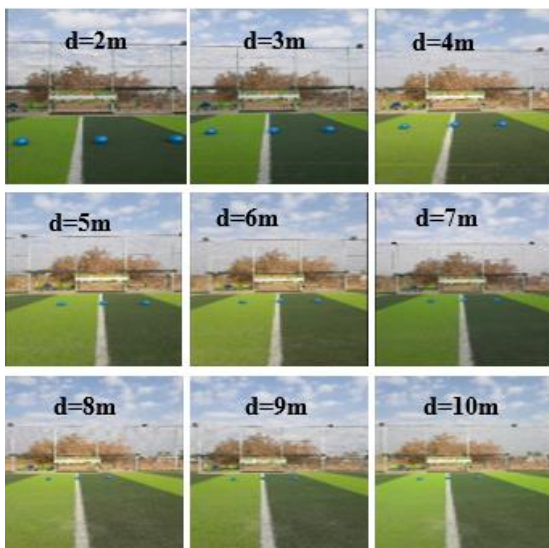


Figure 4. Distance between camera and balls

#### 4. METHODOLOGY

A unique approach constructed with MATLAB R2020a is developed. The cascade classifier is used for training the computer to detect balls in an image and then determine the perfect distance for detecting balls. The introduced approach uses computer training at various distances between the camera and the balls. The computer-training stage and the ball detection stage carried out independently for the objectives of this work.

##### 4.1 Training stage

At this stage, the introduced system model is trained based on a cascade classifier to detect balls in images. Where balls

in any input images or video frames are detected using a ball classifier model that is obtained as a result of training step. The system model has been trained using a boosted algorithm to train 20 levels of a cascading classifier. A degenerated decision tree with levels of increasing complexity is used to train the computer using a cascade of classifiers. At the first level, a classifier is trained to detect nearly all objects of interest (balls), which prompts evaluation of the classifier at the second level, which has also been adjusted to achieve a very high detection rate. A positive result from the second level leads a third level, and so on. A negative result in any level leads to the immediate rejection of the sub-window. A series of classifiers are applied to every sub-window. After several levels of processing, the number of sub-windows reduces radically. The overall form of the detection process is that of a degenerate decision tree, which is called a "cascade" [31]. The AdaBoost method generates a weak classifier "feature" in each cascade level, which is then utilized to choose a single HOG feature. Using the weighted sum of all weak classifiers, a single AdaBoost combines all weak classifiers (weak features) from all levels to produce a single strong classifier (strong feature) [7].

In each level, the HOG feature has been used to classify images, whether they contain a region of interest (ball) or not. At the end of training, the ball classifier model resulted as a file in Extensible Markup Language (XML). The resulting XML-file contains significant features of the ball as a result of training with a cascade object classifier. The MATLAB classifier function *trainCascadeObjectDetector* utilized to train a system using two image datasets: positive images and negative images. The positive image dataset consists of images containing ball objects. While the negative dataset includes images of any object except balls. The block diagram of cascade training is shown in Figure 5.

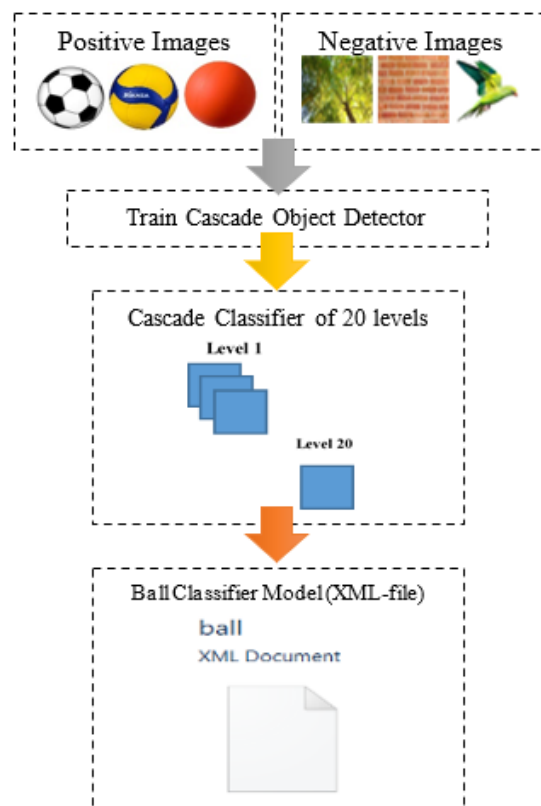


Figure 5. Block diagram of the introduced system model training using cascade classifier

As a pre-training step using a cascade classifier, the region of interest ROI (ball) is labeled in all positive datasets. The ball was labeled manually using the Image Labeler application, which is included in the Computer Vision System Toolbox in MATLAB. To label the ball in positive image data, the ball object in each positive image is enclosed by a bounding box. Each bounding box is configured as a vector of the format [x, y, width, height] to specify the ball location and size. Where x and y specify the ball object location as they represent the coordinates of the top-left corner of the rectangle that contains the ball, while width and height represent its size. After labeling all the positive image datasets, a two-column table was obtained known as the positive samples. In the first column, each row represents a positive image name, whereas the rows of the adjacent column contain the bounding box vector information of the ball of the corresponding positive image. As for the negative samples, they are rejected at each training level.

#### 4.2 Ball detection stage

The new nine video clips (vid1, vid2, vid3, ..., and vid9) that were used as inputs were subjected to the suggested algorithm, whose steps are detailed in Figure 6. This so-called "Ball Detection Algorithm" aims to identify all balls within each video frame and calculate their width and height. This algorithm consists of several steps demonstrated as follow:

**Step1:** Extracting 9 frames from the video input ( $I_f = f_1, f_2, f_3, \dots, f_9$ ).

**Step2:** Resizing the image ( $I_f$ ) by the *imresize* function. Note that the five magnification factors ( $RS = 1, 3, 5, 7, \text{ and } 9$ ) were applied sequentially to all frames extracted from the nine videos.

**Step3:** Detecting balls using a model classifier that saved as an XML-file from training stage.

**Step4:** Cut out each ball detected in the image.

**Step5:** Calculate the width and height of each rectangle surrounded by the detected ball.

**Repeat the steps:** step 2- step 5 using another magnification factor.

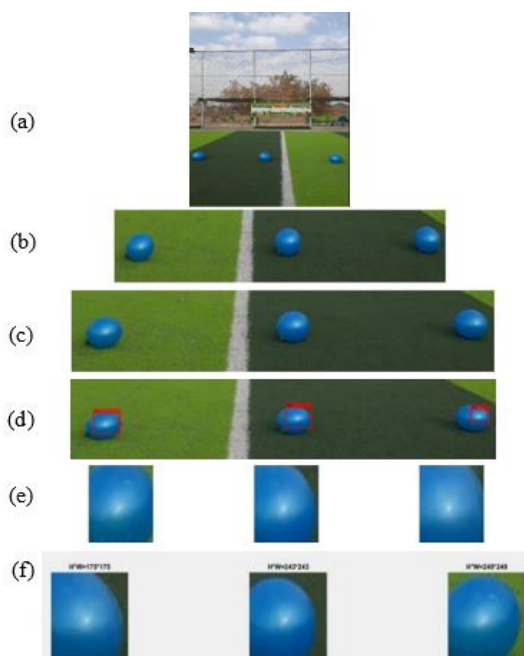


Figure 6. Balls detection steps

After the balls are detected, the image of each ball is extracted, as seen in step 5 of Figure 6. The algorithm then determines the height and width of the extracted images of the detected balls based on their sizes. Where the width and height for the first, second, and third balls are:  $W_1 H_1, W_2 H_2, \text{ and } W_3 H_3$ , respectively, as seen in the above extracted ball image. When applying an XML file on the resized images, it may produce one or both of the following cases:

(1) If all three balls are detected then, they are enclosed by the red rectangle with the name 'Ball', indicating that true positive (TP) detection has occurred. Where the ball considered as true target.

(2) When two, one, or no ball detection occurred, i.e., two, one rectangle-enclosed detected balls, or no rectangle-enclosed target. Thus, false negative (FN) detection results.

### 5. RESULTS AND DISCUSSIONS

In this study, a cascade classifier based on the HOG feature was used to train the computer. The training process was intended to create a strong classifier model that could identify and locate balls. Nine different datasets (video clips) were tested using the suggested procedure. First, nine frames are extracted from each input video:  $f_1, f_2, f_3, \dots, f_9$ . Then, all extracted frames are resized using the following five magnification factors  $RS: 1, 3, 5, 7, \text{ and } 9$ . For each magnification factor, there are nine videos multiplied by nine frames that represent the test data, for a total of  $81 \times 5 = 405$  frames with three blue balls in each frame. Because of exceeding the paper limit, not all results are shown in this research. However, Figures 7 and 8 display the outcomes of the procedure for nine frames extracted from vid2 and vid3, respectively, after resizing the frames by magnification factor  $RS = 9$ . As seen in Figures 7 and 8, the magnification factor  $RS = 9$  detects all balls in all frames. Consequently, true target detection (TP) occurred, and the detection rate is 100%.

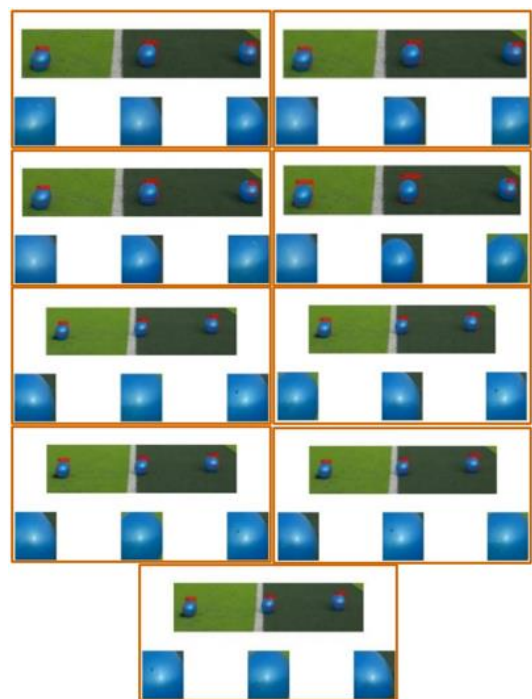
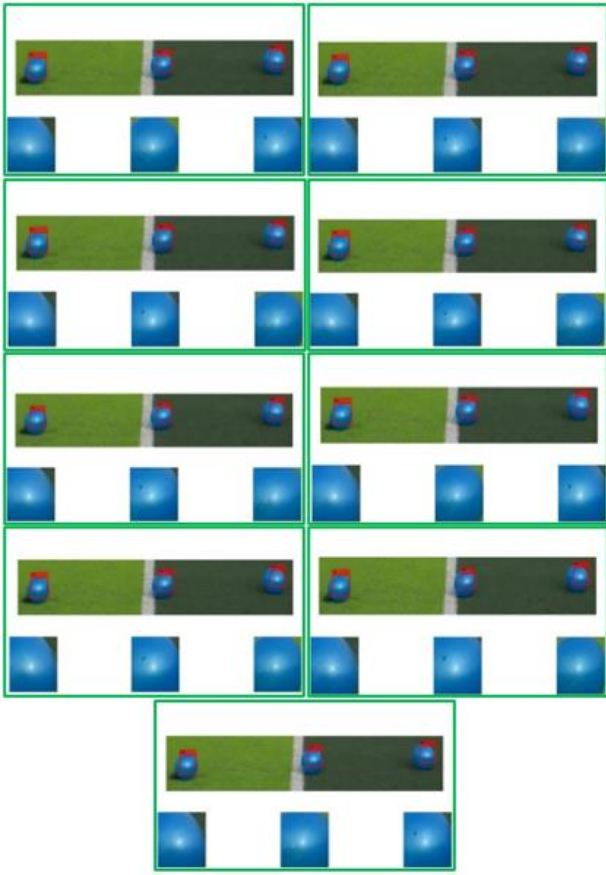
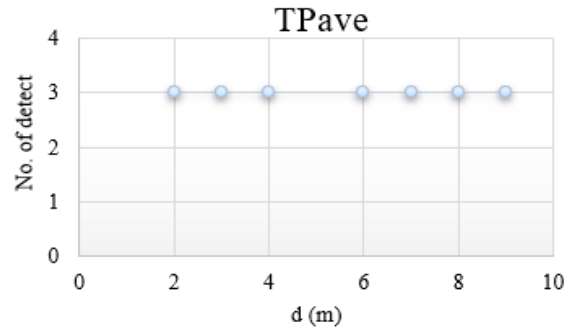
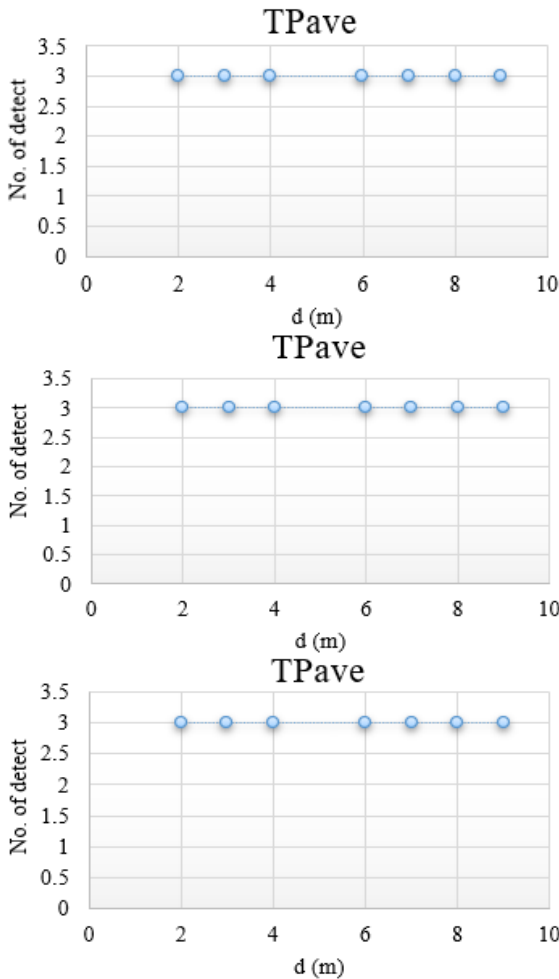


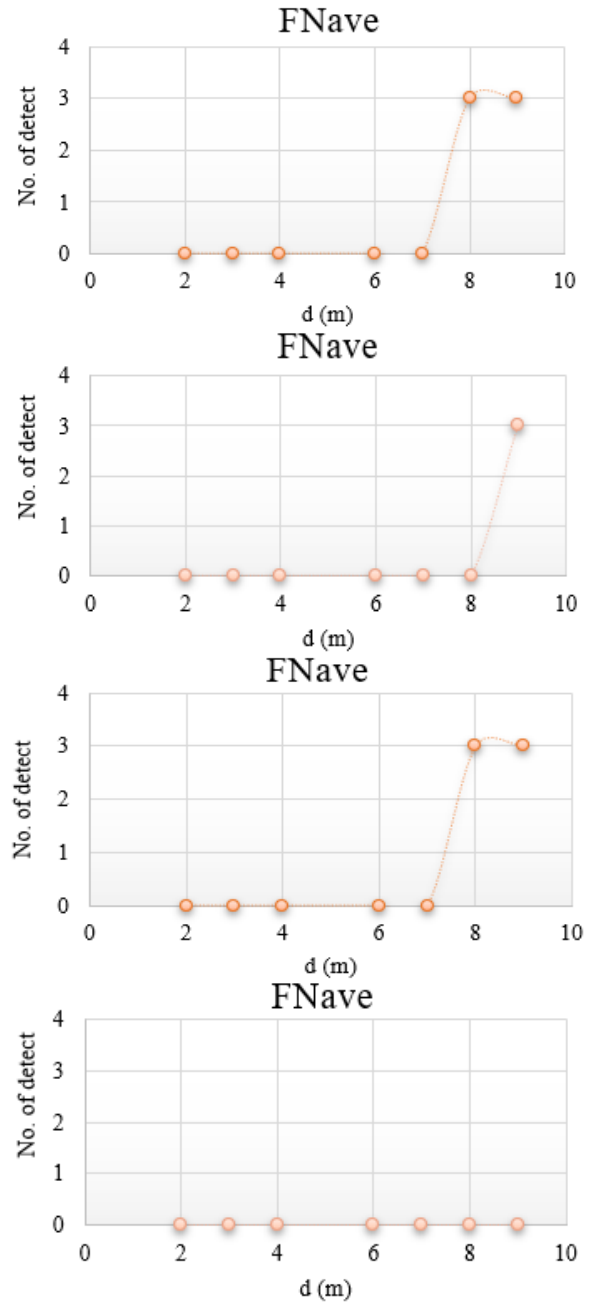
Figure 7. Ball detection for nine frames at a distance of  $d=2m$  between the camera and the balls



**Figure 8.** Ball detection for nine frames at a distance of  $d=3\text{m}$  between the camera and the balls



**Figure 9.** The  $TP_{ave}$  as a function of distance at different RS



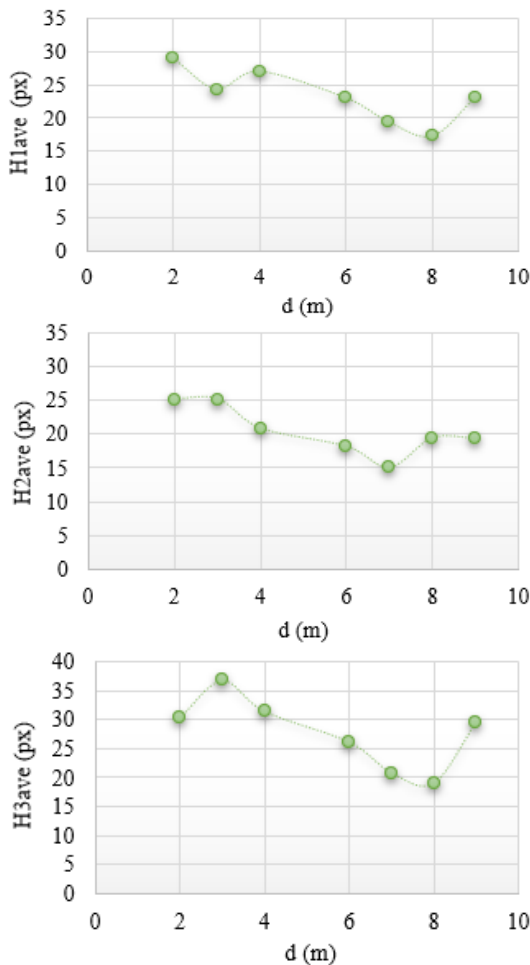
**Figure 10.** The  $FN_{ave}$  as a function of distance at different RS

In general, different outcomes were achieved by employing the proposed magnification factors (RS) on each video. In some videos, not all of the targets (balls) are identified when  $RS = 1$  is used; one target is surrounded by the red rectangle, as a result of which a 20% false target is detected for the balls.

In contrast, findings with  $RS = 3$  and  $5$  for particular videos showed that few targets were not identified in such instances, leading to (FN) detection. The true target (TP) detection rate was  $90\%$  for  $RS = 3$  and  $5$ . While, with  $RS = 9$ , all of the targets were detected. Therefore, all balls are surrounded by a detection rectangle as shown in Figures 9 and 10, which means that  $100\%$  true target detection was achieved.

Anyone may ask a self-evident inquiry regarding the results and say, "When the target is enlarged, it is simpler to detect." However, as the targets' spatial resolution varies with distance, the aim of image resizing is to improve the camera's spatial resolution. Therefore, regardless of whether there is a  $2\text{ m}$  or  $9\text{ m}$  distance between the camera and the balls, all images extracted from the video recordings will be resized.

As well, the height and width of the detected balls have been calculated and then analyzed as a function of the distance ( $d$ ) between the camera and the balls. Where, the average width and height for the first, second, and third balls are ( $W1_{ave}$ ,  $H1_{ave}$ ,  $W2_{ave}$ ,  $H2_{ave}$ , and  $W3_{ave}$ ,  $H3_{ave}$ ), respectively. For a given input video, i.e., at a certain distance ( $d$ ) between the camera and the balls, the  $W$  and  $H$  are computed for the three balls. Next,  $W_{ave}$  and  $H_{ave}$  are calculated that represent the average width and height of balls in each nine extracted frames after resizing by  $RS$  factors, therefore, the result is a single value (average) of width  $W_{ave}$  and height  $H_{ave}$  for each ball versus each distance. The results of the ball height average for first  $H1_{ave}$ , second  $H2_{ave}$ , and third  $H3_{ave}$ , as a function of distance  $d = 2, 3, 4, \dots, 10\text{ m}$ , are illustrated in Figure 11, it can be seen that the height of the ball decreases with increasing distance.



**Figure 11.** The average height of the three balls at different distances

## 6. CONCLUSION

The task of object detection is fundamental in computer vision applications like home automation, surveillance systems, face detection, autonomous driving, and vehicle identification, among others. The ball detection algorithm aims to detect all balls within each video frame and calculate their width and height. Using our approach, we were able to achieve a very suitable ending.

This work introduced a new approach for object detection by resizing the image (multiple magnifications) to preserve the image's high-level features throughout testing. The AdaBoost algorithm was applied through 20 levels of a cascade classifier's training stage. The HOG feature has been employed in each level to categorize images regardless of whether they contain a region of interest (targets) or not. According to the findings, all of the balls had been detected at all distances, and the optimal  $RS$  was  $9$ . When this scale factor is applied, a detection rate of  $100\%$  was achieved. The width and height of the identified balls were determined, and it was found that the width and height of the ball decrease as the distance increases.

## ACKNOWLEDGMENT

The authors would like to express their gratitude to Mustansiriyah University ([www.uomustansiriyah.edu.iq](http://www.uomustansiriyah.edu.iq)) in Baghdad, Iraq, for its assistance with this work.

## REFERENCES

- [1] Xu, P., Zhou, Z., Geng, Z. (2022). Technical research on moving target monitoring and intelligent tracking algorithm based on machine vision. *Wireless Communications and Mobile Computing*, 2022: 7277926. <https://doi.org/10.1155/2022/7277926>
- [2] Taban, D.A., Al-Zuky, A., Al-Saleh, A.H., Mohamad, H. J., Daway, H.G. (2020). ON/OFF switch control of smart home prototype using palm and fist hand gesture. *AIP Conference Proceedings*, 2290(1): 040007. <https://doi.org/10.1063/5.0027540>
- [3] Martínez-Martín, E., del Pobil, A.P. (2013). Robust object recognition in unstructured environments. *Intelligent Autonomous Systems 12*, pp. 705-714. [https://doi.org/10.1007/978-3-642-33926-4\\_67](https://doi.org/10.1007/978-3-642-33926-4_67)
- [4] Ahamad, A.H., Zaini, N., Latip, M.F.A. (2020). Person detection for social distancing and safety violation alert based on segmented ROI. In 2020 10th IEEE International Conference on Control System, Computing and Engineering (ICCSCE), Penang, Malaysia, pp. 113-118. <https://doi.org/10.1109/ICCSCE50387.2020.9204934>
- [5] Deng, J., Xuan, X., Wang, W., Li, Z., Yao, H., Wang, Z. (2020). A review of research on object detection based on deep learning. *Journal of Physics: Conference Series* 1684: 012028. <https://doi.org/10.1088/1742-6596/1684/1/012028>
- [6] Zou, Z., Shi, Z., Guo, Y., Ye, J. (2019). Object detection in 20 years: A survey. *arXiv preprint arXiv:1905.05055*.
- [7] Viola, P. (2001). Rapid object detection using a boosted cascade of simple features. In *IEEE Conference on*

- Computer Vision and Pattern Recognition, Kauai, HI, USA. <https://doi.org/10.1109/CVPR.2001.990517>
- [8] Fauzan, A.M., Ibrahim, S., Sulisty, M.E. (2022). Implementation of object detection method for intelligent surveillance systems at the faculty of engineering, Universitas Sebelas Maret (UNS) Surakarta. *Journal of Electrical, Electronic, Information, and Communication Technology*, 4(1): 24-28. <https://doi.org/10.20961/jeeict.4.1.61197>
- [9] Ingle, P.Y., Kim, Y.G. (2022). Real-time abnormal object detection for video surveillance in smart cities. *Sensors*, 22(10): 3862. <https://doi.org/10.3390/s22103862>
- [10] Khan, S., Nazir, S., Khan, H.U. (2021). Smart object detection and home appliances control system in smart cities. *Computers, Materials and Continua*, 67(1): 895-915. <https://doi.org/10.32604/cmc.2021.013878>
- [11] Wang, D., Wang, J.G., Xu, K. (2021). Deep learning for object detection, classification and tracking in industry applications. *Sensors*, 21(21): 7349. <https://doi.org/10.3390/s21217349>
- [12] Gao, J., Yang, Y., Lin, P., Park, D.S. (2018). Computer vision in healthcare applications. *Journal of Healthcare Engineering*, 2018: 5157020. <https://doi.org/10.1155/2018/5157020>
- [13] Devi, B.T., Shitharth, S. (2021). Multiple face detection using HAAR-AdaBoosting, LBP-AdaBoosting and neural networks. *IOP Conference Series: Materials Science and Engineering*, 1042: 012017. <https://doi.org/10.1088/1757-899X/1042/1/012017>
- [14] Mirza, N.M. (2018). Printed arabic characters recognition based on minimum distance classifier technique. *Iraqi Journal of Science*, 59(2A): 762-770. <https://doi.org/10.24996/ij.s.2018.59.2A.14>
- [15] Sautier, C., Puy, G., Gidaris, S., Boulch, A., Bursuc, A., Marlet, R. (2022). Image-to-lidar self-supervised distillation for autonomous driving data. *arXiv preprint arXiv:2203.16258*.
- [16] Mirza, N., Taban, D., Karam, A., Al-Saleh A., Al-Zuky, A. (2022). Static hand gesture angle recognition via aggregated channel features (ACF) detector. *Traitement du Signal*, 39(3): 939-944. <https://doi.org/10.18280/ts.390320>
- [17] Joseph, S., Pradeep, A. (2017). Object tracking using HOG and SVM. *International Journal of Engineering Trends and Technology (IJETT)*, 48(6): 321-325. <https://doi.org/10.14445/22315381/IJETT-V48P257>
- [18] Shah, S.S.A., Khalil, M.A., Shah, S.I., Khan, U.S. (2018). Ball detection and tracking through image processing using embedded systems. In 2018 IEEE 21st International Multi-Topic Conference (INMIC), Karachi, Pakistan, pp. 1-5. <https://doi.org/10.1109/INMIC.2018.8595582>
- [19] Chowdhury, A.M., Jabin, J., Efaz, E.T., Adnan, M.E., Habib, A.B. (2020). Object detection and classification by cascade object training. In 2020 IEEE International IOT, Electronics and Mechatronics Conference (IEMTRONICS), Vancouver, BC, Canada, pp. 1-5. <https://doi.org/10.1109/IEMTRONICS51293.2020.9216377>
- [20] Sultana, M., Ahmed, T., Chakraborty, P., Khatun, M., Hasan, M.R., Uddin, M.S. (2020). Object detection using template and HOG feature matching. *International Journal of Advanced Computer Science and Applications*, 11(7): 233-238. <https://doi.org/10.14569/IJACSA.2020.0110730>
- [21] Nugraha, A.C., Hakim, M.L., Yatmono, S., Khairudin, M. (2021). Development of ball detection system with YOLOv3 in a humanoid soccer robot. *Journal of Physics: Conference Series*, 2111: 012055. <https://doi.org/10.1088/1742-6596/2111/1/012055>
- [22] Ferreira, A.J., Figueiredo, M.A. (2012). Boosting algorithms: A review of methods, theory, and applications. *Ensemble Machine Learning*, pp. 35-85. [https://doi.org/10.1007/978-1-4419-9326-7\\_2](https://doi.org/10.1007/978-1-4419-9326-7_2)
- [23] Vikram, K., Padmavathi, S. (2017). Facial parts detection using Viola Jones algorithm. In 2017 4th International Conference on Advanced Computing and Communication Systems (ICACCS), Coimbatore, India, pp. 1-4. <https://doi.org/10.1109/ICACCS.2017.8014636>
- [24] Dalal, N., Triggs, B. (2005). Histograms of oriented gradients for human detection. In 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05), San Diego, CA, USA, pp. 886-893. <https://doi.org/10.1109/CVPR.2005.177>
- [25] Fleyeh, H., Roch, J. (2013). Benchmark evaluation of HOG descriptors as features for classification of traffic signs. *Högskolan Dalarna*. [https://doi.org/10.7708/ijtte.2013.3\(4\).08](https://doi.org/10.7708/ijtte.2013.3(4).08)
- [26] Cruz, J., Shiguemori, E., Guimaraes, L. (2015). A comparison of Haar-like, LBP and HOG approaches to concrete and asphalt runway detection in high resolution imagery. *Journal of Computational Interdisciplinary Sciences*, 6(61): 121-136. <https://doi.org/10.6062/jcis.2015.06.01.0101>
- [27] Akila, K., Pavithra, P. (2021). Optimized scale invariant hog descriptors for object and human detection. *IOP Conference Series: Materials Science and Engineering*, 1119: 012002. <https://doi.org/10.1088/1757-899X/1119/1/012002>
- [28] Zhu, Q., Yeh, M.C., Cheng, K.T., Avidan, S. (2006). Fast human detection using a cascade of histograms of oriented gradients. In 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06), New York, NY, USA, pp. 1491-1498. <https://doi.org/10.1109/CVPR.2006.119>
- [29] Jyothi, R., Mahalakshmi, K., Vaishnavi, C.K., Apoorva, U., Nitya, S. (2019). Driver assistance for safe navigation under unstructured traffic environment. In 2019 Global Conference for Advancement in Technology (GCAT), Bangalore, India, pp. 1-5. <https://doi.org/10.1109/GCAT47503.2019.8978279>
- [30] Freund, Y., Schapire, R.E. (1997). A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, 55(1): 119-139. <https://doi.org/10.1006/jcss.1997.1504>
- [31] Viola, P., Jones, M.J., Snow, D. (2005). Detecting pedestrians using patterns of motion and appearance. *International Journal of Computer Vision*, 63(2): 153-161. <https://doi.org/10.1007/s11263-005-6644-8>