Deep Learning Algorithms for Human Fighting Action Recognition

https://doi.org/10.3991/ijoe.v18i02.28019

Mohammed Abduljabbar Ali[1(✉)], Abir Jaafar Hussain[1,2], Ahmed T. Sadiq[1,2]
[1] Computer Sciences Department, University of Technology, Baghdad, Iraq
[2] School of Computer Sciences and Mathematics, Liverpool John Moores University, Liverpool, England
cs.19.21@grad.uotechnology.edu.iq

Abstract—Human action recognition using skeletons has been employed in various applications, including healthcare robots, human-computer interaction, and surveillance systems. Recently, deep learning systems have been used in various applications, such as object classification. In contrast to conventional techniques, one of the most prominent convolutional neural network deep learning algorithms extracts image features from its operations. Machine learning in computer vision applications faces many challenges, including human action recognition in real time. Despite significant improvements, videos are typically shot with at least 24 frames per second, meaning that the fastest classification technologies take time. Object detection algorithms must correctly identify and locate essential items, but they must also be speedy at prediction time to meet the real-time requirements of video processing. The fundamental goal of this research paper is to recognize the real-time state of human fighting to provide security in organizations by discovering and identifying problems through video surveillance. First, the images in the videos are investigated to locate human fight scenes using the YOLOv3 algorithm, which has been updated in this work. Our improvements to the YOLOv3 algorithm allowed us to accelerate the exploration of a group of humans in the images. The center locator feature in this algorithm was adopted as an essential indicator for measuring the safety distance between two persons. If it is less than a specific value specified in the code, they are tracked. Then, a deep sorting algorithm is used to track people. This framework is filtered to process and classify whether these two people continue to exceed the programmatically defined minimum safety distance. Finally, the content of the filter frame is categorized as combat scenes using the OpenPose technology and a trained VGG-16 algorithm, which classifies the situation as walking, hugging, or fighting. A dataset was created to train these algorithms in the three categories of walking, hugging, and fighting. The proposed methodology proved successful, exhibiting a classification accuracy for walking, hugging, and fighting of 95.0%, 87.4%, and 90.1%, respectively.

Keywords—VGG-16, human action recognition, YOLOv3, deep learning, OpenPose

1 Introduction

Recognizing human activities has remained among the most significant problems in computer vision. The demand for human action recognition algorithms has been continuously increasing. These methods have increased in various sectors, such as human-computer interaction, video indexing/retrieval, visual surveillance, video summary, and video understanding [1] [2]. Deep learning-based algorithms for person recognition in computer vision have been widely employed in recent decades. The convolutional neural network (CNN) has become a popular tool for solving real-time challenges [3]. In various robust and discriminative configurations, the CNN has been used for image processing, passenger flow calculations, crowd counting, and object recognition [4]. Human activity recognition on the fly has various challenges, unlike the detection of off-line activities and identification. It is desirable to identify the start and completion points of the action along the time axis and the action type as soon as possible [5][6].

This article outlines a fast method for detecting human fighting inside organizations, which is critical for monitoring to achieve real-time security. The idea is based on selecting a particular frame for the video from a series of frames using the YOLOv3 update to improve the speed of human detection and calculate a safe distance between them to choose a scenario that depicts a fight or handshake. Additionally, OpenPose and a deep sorting algorithm are used for monitoring people when recognizing the instances above, and a pretrained VGG-16 algorithm distinguishes human combat. The results we obtained demonstrate the technology's efficacy. Our contributions in this work are described as follows:

- The collection of primary data consists of three classes: walking, hugging, and fighting. The data consist of various fighting and hugging scenes because they are similar in computer vision.
- The modified You Only Look Once (YOLOv3) algorithm increases the speed of human detection by changing the filter size with the stride.
- A new algorithm is proposed to select a frame from a video based on two deep learning algorithms (modified YOLOv3 and deep simple online and real-time tracking [Deep SORT]) to recognize human fighting in real time.

The remainder of this paper is organized as follows. Section 2 includes the related work, and Section 3 presents the YOLOv3 algorithm. The transfer learning in the deep CNN (DCNN) is illustrated in Section 4. Next, Section 5 discusses proposal work. The experimental results, discussion, and conclusions are provided in Sections 6, 7, and 8, respectively.

2 Related work

There has been a lot of research on existing machine vision-based fall detection systems. Pre-processing stage, feature extraction, classifying, and detection are the three phases that make up the algorithm's flow. The focus of the study has shifted to

the implementation and improvement of the last two steps. This section provides a literature review on human detection, tracking and human action recognition.

2.1 Human detection and tracking

Widespread research has examined developing efficient and reliable object detecting systems. Deep learning, which is unquestionably the current de facto approach for object detection, has been used in various studies [7][8]. Putra et al. offered a real-time human and automobile identification system that may be used in intelligent vehicles or advanced driver assistance systems. The technique is based on a modified YOLO that employs seven layers of CNNs. The system's grid cells were changed to test the efficacy and capacity to recognize tiny objects and automobiles in real-world photos [9]. Ren et al. investigated people counting in real time using video records, a critical component of many innovative city applications. In practice, this activity frequently encounters issues, such as the inability to analyze recorded films in real time or the possibility of mistakes owing to counting irrelevant people. This study offers YOLO-based people counting, a unique real-time people counting technique to address previously mentioned problems [10]. Ahmad et al. proposed detecting people from different angles, namely from above. The deep learning model uses artificial intelligence to learn new things. In-person detection from an overhead perspective, YOLO, has been investigated. The model is evaluated on an overhead view human dataset after being trained on facial image data. Furthermore, overhead view human counting was performed using categorized bounding box information [11]. Lei et al. presented an object tracking approach based on YOLOv3 and the MeanShift method coupled with a Kalman filter to cope with rapidly moving objects and target occlusion to improve the tracking speed and accuracy. This approach employs YOLOv3, tracks it, and uses the MeanShift technique coupled with the Kalman filter to detect the target [12].

2.2 Human action recognition

More deep neural models have been developed to solve the action recognition problem [13][14]. As a result, skeleton information is increasingly being used in human activity recognition. Furthermore, the ability to extract skeleton data in real time using a single RGB camera makes skeleton-based action analysis even more promising. Furthermore, human activities may be recognized directly using skeleton data [15].

Zhang et al. proposed an ergonomic posture identification approach based on three-dimensional (3D) view-invariant characteristics from a single 2D camera that is non-intrusive and extensively used on construction sites. In their approach, a multistage CNN architecture extracted positions relative to the 3D joint positions and joint angle as classification variables based on the detected 2D skeletons. The trained classifier is not sensitive to camera views. Three posture classifiers for the arms, back, and legs were taught to classify them simultaneously in a single video frame [16]. Liu et al. provided a unique and unified framework for skeleton-based posture identification,

using a 3D solid CNN to address the problem. In addition, bounding-box-based normalization for raw skeletal data was presented to remove the coordinate discrepancies induced by different recording conditions and posture displacements [17]. Furthermore, for the skeleton, Gaussian voxelization was used to describe the posture configuration expressively. As a result, 3D Posture Net, an end-to-end framework based on the 3D CNN, was created for comprehensive posture identification. Gatt et al. developed a method for identifying abnormal behavior, such as when a person falls. Pretrained PostNet and OpenPose posture estimate models are used in this work, followed by activity classification using long short-term memory and the CNN [18]. Bulbul et al. illustrated how to differentiate human activities using classification and machine learning techniques, such as bagging, k nearest neighbors, and others. They employed two smartphone sensors, the accelerometer and gyroscope, to do so and were able to detect six different activities [19]. Finally, Zhang et al. presented the "five-point inverted pendulum model," which offers a new human posture representation model for a falling activity that uses an improved two-branch multistage CNN to extract and construct the inverted pendulum structure of the human posture in complex real-world settings [20].

3 YOLOv3

The YOLOv3 (unified, real-time object detection) is a CNN that predicts numerous box locations and classifications at once. It can detect and recognize targets from the beginning to the end. Its most significant benefit is its quickness. Moreover, YOLO is the most recent version of YOLOv3, and it is quicker and more accurate than the previous versions. Figure 1 presents the network architecture of the YOLOv3 algorithm.

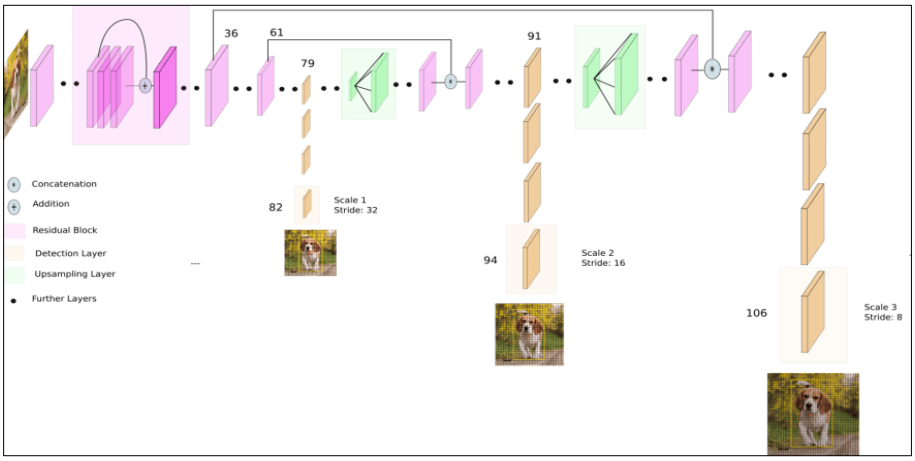

Fig. 1. Network architecture of YOLOv3 [22]

3.1 Bounding box prediction

The YOLOv3 system anticipates bounding boxes, and dimension clusters are used as anchor boxes. Let the four border coordinates be denoted by t_x, t_y, t_w, and t_h. If the cell is offset from the top left corner of the image by (c_x, c_y) and the bounding box has a width and height of p_w and p_h, then the predictions correspond to the following [21]:

$$b_x = \sigma(t_x) + c_x \tag{1}$$

$$b_y = \sigma(t_y) + c_y \tag{2}$$

$$b_w = p_w e^{t_w} \tag{3}$$

$$b_h = p_h e^{t_h} \tag{4}$$

3.2 Class prediction

Instead of softmax, YOLOv3 employs multilabel classification and independent logistic classifiers. Furthermore, during training, it employs binary cross-entropy loss to predict the target class [12].

3.3 Feature extractor

Additionally, YOLOv3 uses Darknet-53 to extract features. Darknet-53 can accomplish the maximum observed floating-point operations per second with fewer floating-point operations, making it more efficient and quicker to evaluate [12].

4 Transfer learning in the DCNN

The DCNN requires large labeled picture datasets [23]. However, in numerous sectors, acquiring and annotating such datasets is difficult and expensive. In the face of such challenges, using "off-the-shelf" properties of well-known DCNNs, such as VGG-16 [24], which have been pretrained on a large-class natural picture dataset, such as ImageNet, has proved effective for picture classification issues through transfer learning [25]. The visualizations at various network layers in the DCNN correspond to many degrees of abstraction in the picture collection [26]. As illustrated in Figure 2, the first convolutional (conv) layer receives an image with a size of 224 × 224 as input. The incoming image is processed through a set of conv layers with a 3 × 3 receptive field. The conv stride is one pixel long. Five max-pooling layers with a stride of two are used for spatial pooling (downsampling). A few conv layers are followed by max-pooling layers applied across a 2 × 2 pixel window. There are entire conv layers after the set of conv layers [24].

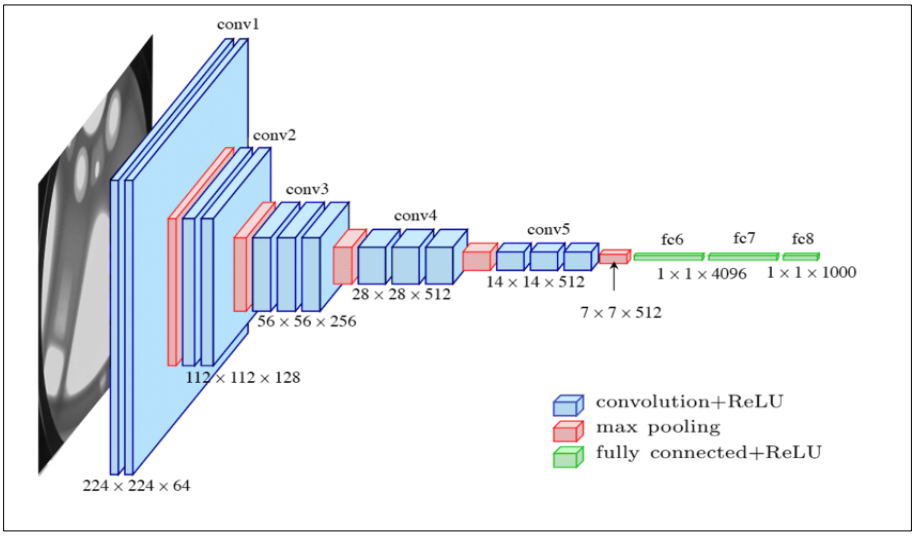

Fig. 2. Basic architecture of VGG-16 [24]

5 Proposed work

This section covers methods for recognizing and estimating human posture in video frames and preprocessing the incoming video comprising noise reduction and frame tuning, as illustrated in Figure 3. For human detection, the number of people and distance between them are computed using the modified YOLOv3 algorithms. The deep SORT using the deep link scale method tracks people if several individuals are in the frame. Partial affinity fields are used to detect crucial points on the human body (human posture), and VGG-16 pretraining is used to predict human actions.

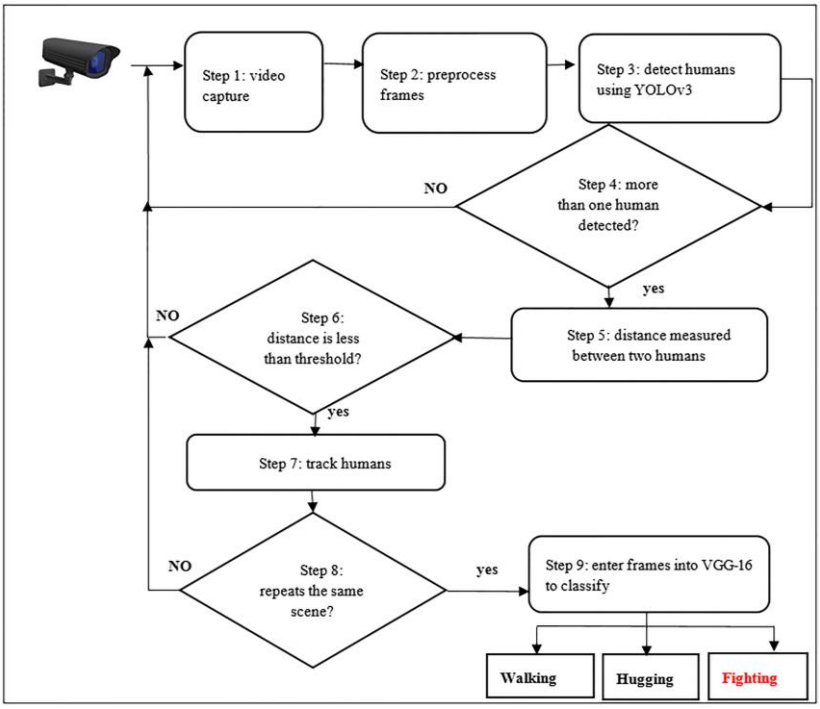

Fig. 3. Approach flowchart

5.1 Data augmentation

Data augmentation aims to produce fresh sample instances, and when the number of training samples is limited, it is a great way to improve network resiliency. Such methods as rotating 30°, 45°, 90°, or 130°, scaling 15% to 40%, chopping, switching the frequency band, vertical or horizontal flipping, and other picture operations are used in this research on remote sensing imaging to increase the network generalization capacity. The validation and testing sets are not affected by the above activities. Rotation and flipping are typically used to expand the number of photographs in the training dataset for the YOLOv3 9000. In rotation-based data augmentation approaches, input pictures are rotated at various angles of 90°, 180°, and 270°. The input pictures are mirrored horizontally and vertically using flipping-based data augmentation methods. Image translations are considered a data augmentation strategy in the literature research, although they provide incorrect results due to their detection and classification task limitations.

5.2 Preprocessing data

The objective of data preprocessing is to extract relevant information to avoid extra data processing costs. The first step is converting the video to frames, then applying a

Gaussian filter to reduce image noise and unwanted information [27].The image is scaled to match the size of the image sent to the YOLOv3 and VGG-16 algorithms.

5.3 Human detection and tracking

Detection. The YOLOv3 algorithm was employed in the human detection proposal, and the leaky activation function was used in the initial YOLOv3 network. A rectified linear unit (ReLU) activation function was constructed, which provides all negative values for a slope that is not zero. Although the usage effect is more significant on a short training set, it is still the most commonly employed ReLU activation function in real-life scenarios. Both functions are "unsaturated activation functions." Griffin Huntington was the first to propose the ReLU function. In matrix x, all negative values are set to zero, but the remainder of the matrix remains intact. The computation occurs after the convolution; therefore, the tan (*h*) and sigmoid functions are used in the same order.

In this work, the YOLOv3 is modified to be faster for detecting humans from original algorithm. The modification is performed by changing the filter size to 7×7 and 5×5 instead of 3×3 and 1×1, and the stride is changed to 3 instead of 2. The size of the human is considered one of the large objects in the image. Hence, the method does not require filters for small sizes that could lead to time-consuming delays in the calculations. Then, the ReLU function to replace the leaky activation function can efficiently decrease the computation, especially when several parameters exist. The modified YOLOv3 algorithm was used to detect humans, and it detects additional objects (80 objects) based on the training COCO dataset. The detection of humans in any frame of a video is illustrated in Figure 4. Human action classification only requires human detection. Therefore, a human being is exposed and ignored by another being. Therefore, the number of human detections is counted. If more than one person is in the video frame, the proposed procedure proceeds to the next phase; otherwise, the frame and any other procedures are skipped. The distance between two persons is measured through the center of the bounding box detected in this stage while assessing whether the frame contains more than one person. Figure 5 demonstrates how the bounding box center is determined for two people near each other.

Fig. 4. Detection of multiple humans using YOLOv3

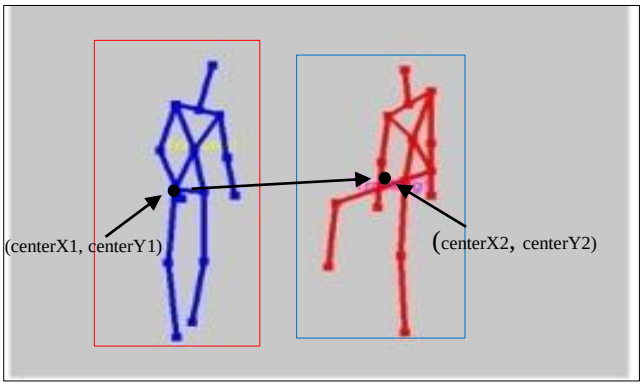

Fig. 5. Measurement distance from the center bounding box between two people

The frame is considered significant by the test distance between two people if it is less than a specific value determined in the code. The people are tracked, and the distance between them is measured again. The method processes the frames in the following stage if the distance in every 120 frames is smaller than the threshold. Thus, it takes less time for the algorithm to categorize human behaviors in the video. Furthermore, by testing this method on the dataset, it works in real time. Fighting between two or more people indoors is one of the most severe threats to the institution's security and must be identified in real time.

Tracking. The deep learning-based tracking algorithm Deep SORT [28] was used for tracking a person from the top perspective, as illustrated in Fig. 6. The two primary components are the frame-by-frame data association technique and Kalman filtering. The filtering evaluates the tracks that already exist in the current video frames. It

uses x', y', h', γ', u, v, h, and γ, where x', y', h', and γ' monitor the velocity of each detected bounding box coordinate and (u, v, h, γ) is the bounding box location [28].

The Kalman filter is used with Deep SORT, linear observations, and the constant velocity. As a result, the position of each existing track in the current frame is predicted in the next frame. The track is estimated using the bounding box spatial information. An appearance descriptor acquires feature extraction for appearance information at each detection and tracking instance. The CNN model is used to train this descriptor. The trained model extracts the features, and a feature vector is created. The vector groups the aspects of the same identity, whereas features with distinct identities are separated [27].

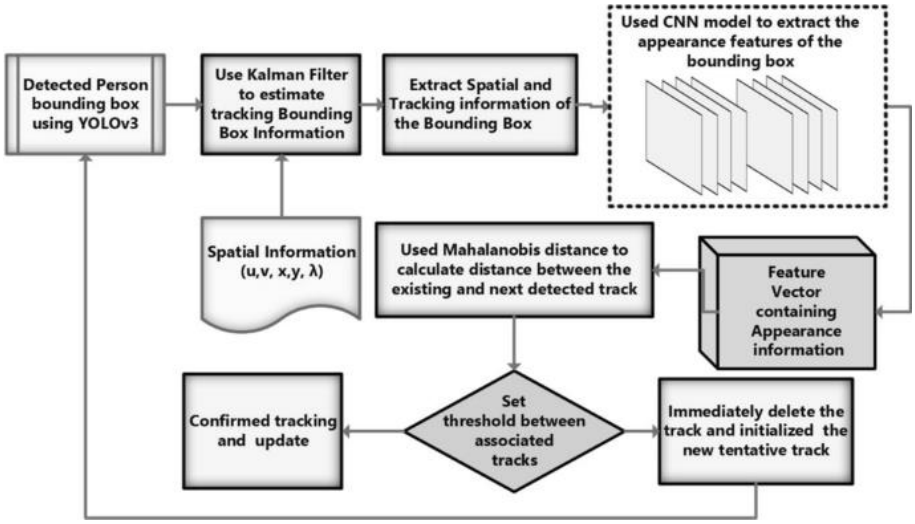

Fig. 6. Top-view person tracking with Deep SORT: A general framework [28]

The new detection results can be connected with the current tracking results in the subsequent frame using the information extracted from the appearance descriptor. A detection threshold is set for this reason, ensuring that the low detection results are ignored. Each detection result is now linked to a threshold in the next frame. The Deep SORT method uses a cost matrix to describe the appearance and spatial similarities between new detections and tracks objects/people using two distance values, represented as follows [28]:

$$d^1(i,j) = \left(d_j - y_i\right)^T S_i^- 1\left(d_j - y_i\right). \tag{5}$$

In the equation above, y_i and S_i symbolize the i_{th} the measuring space in the projection track, and for the j_{th} new detection, d_j is used. This is also known as the Mahalanobis distance, which is the calculated difference between the new detection j_{th} and the estimated position i_{th} track. Furthermore, the Mahalanobis distance threshold between improbable connections is eliminated using the measure mentioned above the j_{th} detection and the i_{th} track, as given in [28]:

$$b_{i,j}^{1} = 1\left[d_{(i,j)} < t\right]. \tag{6}$$

The following equation is used to estimate the second distance value, which represents the appearance information [28]. The lowest cosine distance between the j_{th} detection and the i_{th} track was determined using this second distance value as follows:

$$d^{2}(i,j) = \min\left(1 - r_j^T r_k^{(i)} \middle| r_k^{(i)} E R_i\right). \tag{7}$$

The appearance descriptor is r in the above equation, and R_i describes the appearance of at least 100 items (people) in the i_{th} track. To set the threshold between the association tracks, we used the following [28]:

$$b_{i,j}^{1} = 1\left[d_{(i,j)} < t\right]. \tag{8}$$

If the distance value is small, it equals 1, and 0 is significant. For more details, we refer to readers to [28]. We estimated the cost function using the following matrix:

$$c_{i,j} = \lambda d_{(i,j)} + (1 - \lambda) d_{(I,j)}. \tag{9}$$

The gate function is given as follows [28] to match the spatial information:

$$b_{i,j} = \prod_{m=1}^{2} b_{i,j}^{m}. \tag{10}$$

If the value of the above equation is 1, the appearance and spatial gate functions are equal, and the value is 0 if they are not. It also suggests that (i, j) is a genuine match between appearance and spatial data. As a result, the detections and tracking in each new video frame are connected with the above cost and gate functions. For processing, tracking in the video sequence continues in the next new video frame when the new detection is effectively connected to the current track. It is set to zero if it is not connected or matched. Thus, the new detections fail in such a scenario. When new detections fail to correlate with existing detections in frame f, the new detections are started as tentative tracks. The Deep SORT algorithm validates and associates additional detections in subsequent (f + 1), (f + 2), ..., (f + t) tentative frames. That track is confirmed for tracking and updated as long as it is effectively linked. Otherwise, it is removed right away [27].

5.4 Human action classification

The proposed method for detecting the 2D stance of many individuals in a picture was demonstrated. The method employs a nonparametric representation known as part affinity fields to link body parts with people in the picture. The design stores the global context, enabling a greedy bottom-up parsing phase to provide real-time speed while maintaining excellent accuracy, regardless of the number of individuals in the image. Through two branches of the same sequential prediction process, the architecture is meant to learn the part locations and their associations jointly [32]. The characteristics of the earlier layers of a pretrained CNN frequently include edge and color

information. However, the subsequent layers have properties more relevant to the class features. The parameters of the last layers require little or no fine-tuning [23]. Only the last three layers of the VGG-16 were fine-tuned in this study. In addition, VGG-16 was trained on over a million images, and it can classify photographs into 1000 different categories [24]. These 1000 classes are configured in the last three tiers of VGG-16, which need to be fine-tuned for a newer categorization task [29-31]. All layers except the last three are extracted to fine-tune the network. The previous three layers are replaced with an *fc* layer, softmax layer, and classification output layer to move the layers to the new classification task. The size value for this work is three, corresponding to the number of classes, such as walking, embracing, and fighting.

6 Experimental results

6.1 Data description

The evaluation and training of the proposed model were based on a dataset collected in an indoor establishment. The RGB dataset was collected using a single Panasonic HC-MDH2 AVCHD camera, recorded in full high definition (1920 x 1080p). A dataset to identify the situation was created that simulates reality. The camera was installed on a tripod in an enterprise building in our trial setup. Figure 7 illustrates the three types of postures: walking, hugging, and fighting. Two individuals of varying ages and physical features completed each stance. Each posture was collected at three distinct orientations and distances from the camera to improve the variety of the dataset and assess the capacity of the system to handle size and orientation changes. The distance ranges from 1 to 5 m, with an orientation angle of 0° to 360°. The dataset contains 6558 observations for training and testing the suggested system. Table 1 explains each class account. The dataset was divided into two subsets for both approaches, with 80% used for training the systems and 20% for testing.

(a) (b) (c)

Fig. 7. (a) Walking, (b) fighting, and (c) hugging

Table 1. Account for each class in the videos

Categories	Number of frames	Size of video
Walking	2226	157.140 KB
Fighting	3819	231.624 KB
Hugging	513	31.326 KB

6.2 Results

The proposed technique was tested on datasets acquired by a video camera in a building, with postures calculated from the RGB videos. Confusion matrices were used to examine the proposed algorithm performance using the predicted postures from our dataset. This assessment tool involves comparing reference postures to common postures. The genuine classes are represented by the rows of matrices, whereas the columns represent the anticipated classes. As indicated in Table 2, practically all classes had strong recognition scores. The walking courses had a 95% accuracy rate in the setting. However, a significant misunderstanding occurred between the hugging and fighting classes, with accuracy rates of 87.4% and 90.1% for the RGB photos, respectively. The visual resemblance in silhouette appearance between the hugging and fighting stances explains this misclassification.

Table 2. Confusion matrix of human action recognition

Postures	Walking	Hugging	Fighting	total
Walking	2114	45	67	2226
Hugging	0	3338	481	3819
Fighting	13	38	462	513

This section of the results reveals the speed and accuracy of the YOLOv3 modifications performed during this research. In the case of processing a single image, as depicted in Figure 8 and Table 3, when the activation function changes, both accuracy and speed are affected. The amount of processing may be lowered and the detection speed raised by increasing the filter size and stride because we aim to improve YOLOv3 to improve the classification time. The final comparison of accuracy with speed is primarily compared using the original method and modifications of the activation function. The improved YOLOv3 reduced the accuracy by 4% while increasing its speed by around 16%. Thus, the YOLOv3 modification was successful. It is used in specific applications, especially those containing large and transparent objects, by taking advantage of its speed in identifying objects in real time.

Table 3. Time and accuracy of different neural networks under the Darknet network (time units: seconds)

Algorithms	Accuracy	Time Cost
Original YOLOv3	0.99	0.509
YOLOv3 + tan() activation	0.93	0.60.2
YOLOv3 + softmax() activation	0.94	0.583
YOLOv3 + ReLU() activation	0.95	0.471
YOLOv3 our modified	0.95	0.349

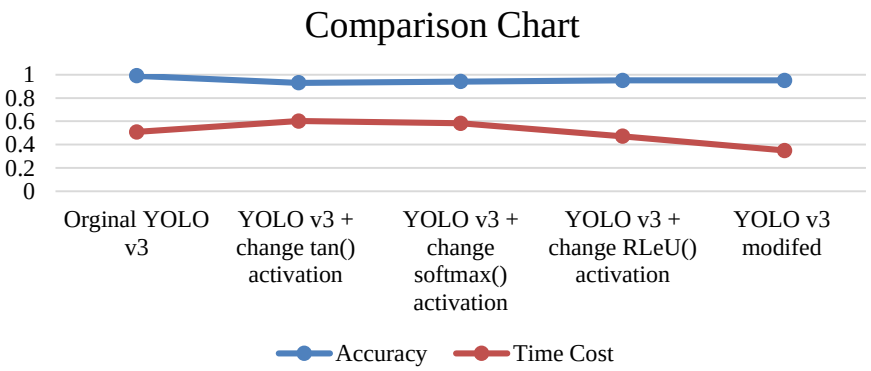

Fig. 8. Timing and accuracy comparison of various activation functions in the Darknet network (time units: seconds)

7 Discussion

The proposed technique could be applied in practice while fulfilling real-time requirements due to high precision and low processing complexity. Many design alternatives are dependent on unique use cases and environmental restrictions. Real-time human fighting action recognition has been fraught with issues. The complicated procedure comprises attribute extraction, data collection, hardware, and classification algorithms. The data collection used to train any model either does not contain all occurrences or is insufficient. The dataset used in this study simulates a portion of human fighting action recognition, on which this study is based. Through training, this data collection was demonstrated to detect fights effectively between people in buildings. Human recognition was sped up by tweaking the YOLOv3 algorithm and choosing a specified number of movie frames. Many applications have been used to classify human activities using the skeleton approach with pretraining with the VGG-16 algorithm.

8 Conclusion

This paper presented new solutions in real time for automatic human fighting action recognition using an RGB camera. This proposal designed a method using visual data provided by an RGB camera. The approach uses YOLOv3 and Deep SORT to detection and track people and choose several frames with critical situations. The VGG-16 algorithm with OpenPose was employed to categorize 2D photos. The suggested approaches were tested using a dataset on fight scenario recognition inside building. The approach performed similarly in the real case to fight, it had a good level of accuracy.

Furthermore, the technique demonstrated a high level of stability for key perturbation elements, such as size and orientation changes. This paper provides many approaches for creating effective situation recognition systems depending on the use cases and application limits. The methods that use RGB pictures, for example, are suitable for indoor real-time situations, and the poses may be easily identified in such an environment, independent of the lighting conditions.

9 References

[1] P. Zhang, C. Lan, J. Xing, W. Zeng, J. Xue, and N. Zheng, "View Adaptive Neural Networks for High Performance Skeleton-Based Human Action Recognition," IEEE Trans. Pattern Anal. Mach. Intell., vol. 41, no. 8, pp. 1963–1978, 2019. https://doi.org/10.1109/tpami.2019.2896631

[2] A. T. Saadeq, E. K. Jabbar, and N. J. Ibrahim, "Human detection and Recognition System," vol. 5, no. 1, pp. 28–51, 2015.

[3] W. M. Salih Abedi, I. Nadher, and A. T. Sadiq, "Modification of deep learning technique for face expressions and body postures recognitions," Int. J. Adv. Sci. Technol., vol. 29, no. 3 Special Issue, pp. 313–320, 2020.

[4] M. Al-Smadi, M. Hammad, Q. B. Baker, and S. A. Al-Zboon, "A transfer learning with deep neural network approach for diabetic retinopathy classification," Int. J. Electr. Comput. Eng., vol. 11, no. 4, pp. 3492–3501, 2021. http://doi.org/10.11591/ijece.v11i4.pp3492-3501

[5] Y. Li, C. Lan, J. Xing, W. Zeng, C. Yuan, and J. Liu, "Online human action detection using joint classification-regression recurrent neural networks," Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics), vol. 9911 LNCS, pp. 203–220, 2016. https://doi.org/10.1007/978-3-319-46478-7_13

[6] N. A. Jasim and H. T. S. ALRikabi, "Design and Implementation of Smart City Applications Based on the Internet of Things," Int. J. Interact. Mob. Technol., vol. 15, no. 13, pp. 4–15, 2021. https://doi.org/10.3991/ijim.v15i13.22331

[7] Z. Cao, T. Liao, W. Song, Z. Chen, and C. Li, "Detecting the shuttlecock for a badminton robot: A YOLO based approach," Expert Syst. Appl., vol. 164, no. July 2020, p. 113833, 2021. https://doi.org/10.1016/j.eswa.2020.113833

[8] F. S. Hameed, Hasan M. Alwan, and Qasim A. Ateia, "Pose Estimation of Objects Using Digital Image Processing for Pick-and-Place Applications of Robotic Arms," Eng. Technol. J., vol. 38, no. 5A, pp. 707–718, May 2020. http://dx.doi.org/10.30684/etj.v38i5A.518

[9] M. H. Putra, Z. M. Yussof, K. C. Lim, and S. I. Salim, "Convolutional neural network for person and car detection using YOLO framework," J. Telecommun. Electron. Comput. Eng., vol. 10, no. 1–7, pp. 67–71, 2018.
[10] P. Ren, L. Wang, W. Fang, S. Song, and S. Djahel, "A novel squeeze YOLO-based real-time people counting approach," Int. J. Bio-Inspired Comput., vol. 16, no. 2, pp. 94–101, 2020. https://doi.org/10.1504/ijbic.2020.109674
[11] M. Ahmad, I. Ahmed, and A. Adnan, "Overhead View Person Detection Using YOLO," 2019 IEEE 10th Annu. Ubiquitous Comput. Electron. Mob. Commun. Conf. UEMCON 2019, no. i, pp. 0627–0633, 2019. https://doi.org/10.1109/UEMCON47517.2019.8992980
[12] W. Lei, D. Huang, and X. Cui, "Moving object tracking in video surveillance using YOLOv3 and MeanShift," vol. 1106940, no. May 2019, p. 86, 2019. https://doi.org/10.1117/12.2524252
[13] Z. Yang, Y. Li, J. Yang, and J. Luo, "Action Recognition with Spatio-Temporal Visual Attention on Skeleton Image Sequences," IEEE Trans. Circuits Syst. Video Technol., vol. PP, no. c, p. 1, 2018. https://doi.org/10.1109/tcsvt.2018.2864148
[14] S. M. Najeeb, S.M. Ali, and H. Salim "Finding the discriminative frequencies of motor electroencephalography signal using genetic algorithm," TELKOMNIKA, vol. 19, no. 1, pp. 285-291, 2021. https://doi.org/10.12928/telkomnika.v19i1.17884
[15] J. Liu, N. Akhtar, and A. Mian, "Skepxels: Spatio-temporal Image Representation of Human Skeleton Joints for Action Recognition," pp. 10–19, 2017, [Online]. Available: http://arxiv.org/abs/1711.05941
[16] H. Zhang, X. Yan, and H. Li, "Ergonomic posture recognition using 3D view-invariant features from single ordinary camera," Autom. Constr., vol. 94, pp. 1–10, Oct. 2018. https://doi.org/10.1016/j.autcon.2018.05.033
[17] J. Liu, Y. Wang, Y. Liu, S. Xiang, and C. Pan, "3D PostureNet: A unified framework for skeleton-based posture recognition," Pattern Recognit. Lett., vol. 140, pp. 143–149, Dec. 2020. https://doi.org/10.1016/j.patrec.2020.09.029
[18] T. Gatt, D. Seychell, and A. Dingli, "Detecting human abnormal behaviour through a video generated model," Int. Symp. Image Signal Process. Anal. ISPA, vol. 2019-Septe, pp. 264–270, 2019. https://doi.org/10.1109/ISPA.2019.8868795
[19] E. Bulbul, A. Cetin, and I. A. Dogru, "Human Activity Recognition Using Smartphones," ISMSIT 2018 - 2nd Int. Symp. Multidiscip. Stud. Innov. Technol. Proc., no. March 2019, 2018. https://doi.org/10.1109/ISMSIT.2018.8567275
[20] J. Zhang, C. Wu, and Y. Wang, "Human fall detection based on body posture spatio-temporal evolution," Sensors (Switzerland), vol. 20, no. 3, p. 946, Feb. 2020. https://doi.org/10.3390/s20030946
[21] J. Redmon and A. Farhadi, "YOLOv3: An Incremental Improvement," 2018, [Online]. Available: http://arxiv.org/abs/1804.02767
[22] L. Miguel and S. Ramos, "Tracking and Counting People with Dynamic Bandwidth Management," theses , 2021.
[23] P. Patel and A. Thakkar, "The upsurge of deep learning for computer vision applications," Int. J. Electr. Comput. Eng., vol. 10, no. 1, pp. 538–548, 2020. https://doi.org/10.11591/ijece.v10i1.pp538-548
[24] T. Kaur and T. K. Gandhi, "Automated brain image classification based on VGG-16 and transfer learning," Proc. - 2019 Int. Conf. Inf. Technol. ICIT 2019, pp. 94–98, 2019. https://doi.org/10.1109/ICIT48102.2019.00023
[25] H. C. Shin et al., "Deep Convolutional Neural Networks for Computer-Aided Detection: CNN Architectures, Dataset Characteristics and Transfer Learning," IEEE Trans. Med. Imaging, vol. 35, no. 5, pp. 1285–1298, 2016. https://doi.org/10.1109/tmi.2016.2528162

[26] M. Zhou et al., "Epileptic seizure detection based on EEG signals and CNN," Front. Neuroinform., vol. 12, no. December, pp. 1–14, 2018. https://doi.org/10.3389/fninf.2018.00095

[27] I. Ahmed, M. Ahmad, A. Ahmad, and G. Jeon, "Top view multiple people tracking by detection using deep SORT and YOLOv3 with transfer learning: within 5G infrastructure," Int. J. Mach. Learn. Cybern., no. 0123456789, 2020. https://doi.org/10.1007/s13042-020-01220-5

[28] N. Wojke, A. Bewley, and D. Paulus, "Simple online and realtime tracking with a deep association metric," Proc. - Int. Conf. Image Process. ICIP, vol. 2017-Septe, pp. 3645–3649, 2018. https://doi.org/10.1109/icip.2017.8296962

[29] P. K. Sonawane and S. Shelke, "Handwritten Devanagari Character Classification using Deep Learning.," 2018 Int. Conf. Information, Commun. Eng. Technol. ICICET 2018, pp. 1–4, 2018. https://doi.org/10.1109/ICICET.2018.8533703

[30] A. Al-zubidi, R. K. Hasoun, and H. Alrikabi, "Mobile Application to Detect Covid-19 pandemic by using Classification Techniques: Proposed System," International Journal of Interactive Mobile Technologies, vol. 15, no. 16, pp. 34-51, 2021. https://doi.org/10.3991/ijim.v15i16.24195

10 Authors

Mohammed Abduljabbar Ali received the BSc and MSc degrees in Computer Sciences in 2003 and 2017, respectively from the University of technology. his research interests are security of data, Encryption algorithms, computer vision and machine learning. (Department Computer Sciences, University of Technology, Baghdad, Iraq).

Ahmed T. Sadiq is a Professor in the Computer Science Department-University of Technology-Iraq. He received a B.Sc., M.Sc. & Ph. D. degree in Computer Science from the University of Technology, Computer Science Department, Iraq, 1993, 1996 & 2000 respectively. He is Professor in A.I. since 2014. His research interests in Artificial intelligence, data security, patterns recognition & data mining.

Abir Hussain is a professor of Biomedical Science and a member of the Applied Computing Research Group at the Faculty of Engineering and Technology. She completed her PhD study at The University of Manchester (UMIST), UK in 2000 with a thesis title Polynomial Neural Networks for Image and Signal Processing. She has published numerous referred research papers in conferences and Journal in the research areas of Neural Networks, Signal Prediction, Telecommunication Fraud Detection and Image Compression. She has worked with higher order and recurrent neural networks and their applications to e-health and medical image compression techniques. She has developed with her research students a number of recurrent neural network architectures. Her research has been published in a number of high esteemed and high impact journals such as the Expert Systems with Applications, PloS ONE, Electronic Letters, Neuro computing, and Neural Networks and Applications.

Article submitted 2021-11-05. Resubmitted 2021-12-19. Final acceptance 2021-12-20. Final version published as submitted by the authors.