Expert Systems With Applications 205 (2022) 117730

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa

A Faster R-CNN and recurrent neural network based approach of gait recognition with and without carried objects

Rajib Ghosh

Department of CSE, National Institute of Technology Patna, India

ARTICLE INFO

Keywords:
Gait recognition
With and without CO
Faster R-CNN
RNN
LSTM
BLSTM

ABSTRACT

Gait recognition is the identification of any person from his/her walking pattern. Walking pattern of each individual is unique and cannot be replicated by others. But, gait recognition is very difficult if any object is carried by any individual. This article proposes a novel computer vision based method of gait recognition both with and without carried objects (COs) using Faster region convolutional neural network (R-CNN) based architecture. To the best of my knowledge, this is the first investigation based on faster R-CNN for gait recognition in the literature. The Faster R-CNN detects and extracts the pedestrian only in all the frames of the video irrespective of the pedestrian is carrying any object or not. Deep convolutional layers have then been used to generate the feature vector from the walking pattern of the pedestrian generated comprising all the frames. The generated feature vectors from various walking patterns have then been studied using two different versions of recurrent neural network (RNN) namely, long–short term memory (LSTM) and bidirectional long–short term memory (BLSTM) to recognize the walking patterns. To the best of my knowledge, the present investigation uses the BLSTM variant of RNN classifier to recognize the walking patterns for the first time in the literature. The performance of the proposed system has been tested on four widely used public datasets—OU-ISIR Large Population Gait database with real-life carried object (OU-LP-Bag), the OU-ISIR Gait database Treadmill dataset B (OUTD-B), OU-ISIR Large Population Gait database with Age (OULP-Age), and CASIA Gait database B (CASIA-B). The experimental results demonstrate that the proposed gait recognition system outperforms the existing state-of-the-art results.

1. Introduction

Each individual has a unique walking pattern or gait. Gait is the most important biometric feature as it can be figure out at the longest distance without the help of the subject. Gait is generated unconsciously by an individual and so it is almost impossible to replicate by others. These characteristics of gait have led to several investigations (Ben et al., 2019; Li, Makihara, Xu, Yagi, & Ren, 2020; Takemura, Makihara, Muramatsu, Echigo, & Yagi, 2019) on person identification through gait recognition in the recent years. These systems have many applications in the real life, such as surveillance systems, forensics, and criminal investigation (Bouchrika, Goffredo, Carter, & Nixon, 2011; Iwama, Muramatsu, Makihara, & Yagi, 2013; Lynnerup & Larsen, 2014). But, this identification task poses a huge challenge for any computer vision based system due to the following factors:

1. Carrying of any object by the pedestrian: In this situation, the object characteristics will be included in the feature vector of walking style.
2. Presence of any moving object in the background: In this situation, it is very tough to segment the image into foreground and background with foreground showing only a certain pose of walking activity.
3. View angle: The front view of walking activity poses more challenge for the system to discriminate the walking styles from each other in comparison to the side view of the activity.

Although several investigations on computer vision based gait recognition systems have been reported in the literature, very few studies (Li, Makihara, Xu, Yagi, & Ren, 2019; Li et al., 2020) have addressed the problem of carried objects (COs) exists in this research domain. The existing methods capable to overcome this problem can be divided mainly into two categories—discriminative and generative. Discriminative methods have relied on either traditional metric learning techniques (Bashir, Xiang, & Gong, 2010; Guan, Li, & Roli, 2015; Han & Bhanu, 2006; Makihara, Suzuki, Muramatsu, Li, & Yagi, 2017; Xu et al., 2006) or deep convolutional neural networks (DCNNs) (Takemura et al., 2019; Wu, Huang, Wang, Wang, & Tan, 2017; Zhang, Huang, Wang and Yu, 2019) to extract the discriminant features from each walking pattern. Traditional metric learning techniques mainly

E-mail address: rajib.ghosh@nitp.ac.in.

https://doi.org/10.1016/j.eswa.2022.117730
Received 29 July 2021; Received in revised form 10 May 2022; Accepted 31 May 2022
Available online 7 June 2022

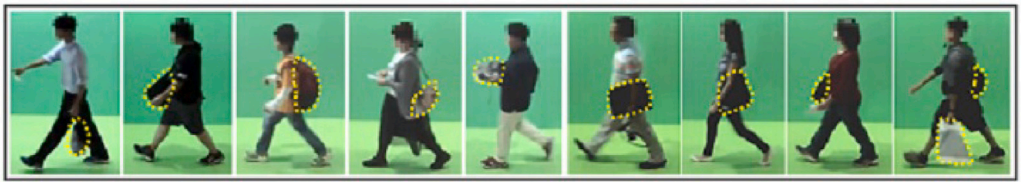

Fig. 1. Walking activity images of different persons with various types of COs with varying shapes, sizes, and locations.

involve spatial metric learning techniques like linear discriminant analysis (LDA) (Han & Bhanu, 2006), discriminant analysis with tensor representation (DATER) (Xu et al., 2006), the random subspace method (RSM) (Guan et al., 2015), intensity transformation based techniques like masked gait energy images (GEIs) (Bashir et al., 2010), gait energy response functions (GERFs) (Li et al., 2016), and a hybrid of intensity and spatial metric learning based techniques (Makihara et al., 2017). DCNN based methods have extracted deep invariant features. However, discriminative approaches cannot recognize a walking pattern correctly if the person in walking carries varying COs because these approaches learn discriminant features of each walking pattern with a few specific COs. Fig. 1 shows walking activity images of different persons with various types of COs. So, it is required to remove the COs from the input images before applying the discriminative approaches to recognize the gait.

On the other hand, studies on generative approaches are limited and they have focused on removing COs from the input images using conventional detection and elimination strategies (Whytock, Belyaev, & Robertson, 2015) or deep generation networks like generative adversarial network (GAN) (He, Zhang, Shan, & Wang, 2019; Li et al., 2020; Yu et al., 2019). However, the existing generative approaches have various limitations like the conventional approach (Whytock et al., 2015) can deal only with the particular type of COs. It cannot deal with the situation if a person carries various types of COs like backpack, drawstring bag, satchel backpack, shoulder bag, waist bag, handheld bag, suitcase, briefcase, trolley bag, etc. Similarly, the GAN based approaches (Yu et al., 2019) successfully remove the COs from the input images but these approaches regenerate other non-CO parts (for example, leg portions of the person carrying a drawstring bag) as well that may lead to unnecessary errors.

So, the present investigation has considered the walking patterns both with and without COs and this article proposes a novel method using Faster R-CNN to detect and extract the pedestrian without CO (in case the pedestrian is carrying any CO) from the input image. No research work has been found in the literature to remove the COs from the input images using Faster R-CNN and the present investigation uses it for the first time in the literature. The Faster R-CNN is a deep learning architecture and was first published in NIPS 2015 (Ren, He, Girshick, & Sun, 2015) to detect distinct objects in scene images. Faster R-CNN provides very high object detection accuracy in comparison to other convolution neural network (CNN) based architectures due to the generation of various region proposals using the region proposal algorithm. The processing time of Faster R-CNN is also lesser in comparison to other CNN based architectures except the architectures use a one-stage method such as YOLOv3 (Redmon & Farhadi, 2018). But, YOLOv3 could not show superior performance than Faster R-CNN in detecting the pedestrian from the GEIs after carrying out the detection experiments with both Faster R-CNN and YOLOv3 in this work (Section 5.3 may be seen). The conventional Faster R-CNN produces regions-of-interest (ROIs) through a single region proposal network (RPN) using the feature matrix generated from the last convolutional layer. Deep convolutional layers have then been used to generate the feature vector from the walking pattern of the detected person. The generated feature vectors from various walking patterns have then been studied using long–short term memory (LSTM) (Ghosh, Vamshi, & Kumar, 2019) and bidirectional long–short term memory (BLSTM) (Ghosh et al., 2019) models of recurrent neural network (RNN) to recognize each walking pattern. RNN can remember the sequential data through these two models for a long duration. Apart from this, BLSTM remembers a sequence in both the directions. The video of walking activity of any individual is a collection of several frames in a sequential order. These constituent frames of the video generate the entire walking pattern. By exploiting the powers of these two variants of RNN, the proposed system recognizes any walking pattern by remembering the sequential order of the concerned frames. Other non-deep machine learning methods do not have these powers and due to which the proposed system employing RNN based deep learning model provides superior gait recognition performances (Section 5 may be seen) in comparison to various non-deep machine learning based methods.

The major contributions of this work are as follows:

1. Use of Faster R-CNN to detect and extract the person only and not the CO from the input image.
2. The use of BLSTM variant of RNN classifier to recognize the walking patterns of different individuals.
3. The use of four widely used public datasets to evaluate the performance of the proposed system.

The rest of the paper is organized as follows. Related works are discussed in Section 2. The details of various datasets are discussed in Section 3. Section 4 details the proposed method of gait recognition. Section 5 analyses the person detection and gait recognition results of the present system. Finally, Section 6 concludes the paper with a direction for future research.

2. Related work

As mentioned earlier, several studies on machine vision based gait recognition systems are available in the literature. Some of those related studies are discussed below in brief.

2.1. Studies using discriminative approaches

Discriminative approaches have extracted various discriminant features from each walking pattern and relied on traditional metric learning and deep learning based methods.

In traditional metric learning based methods researchers have used mainly the following three methods: (1) spatial metric learning based methods, (2) intensity transformation based methods, and (3) a combination of intensity and spatial metric learning based methods. Spatial metric learning based methods have learnt discriminant spatial features from different walking patterns. Han and Bhanu (2006) proposed one spatial metric learning based method using LDA for gait recognition. Xu et al. (2006) proposed another spatial metric learning based method for gait recognition using DATER. Guan et al. (2015) presented another spatial metric learning based method using RSM to recognize the gait. Intensity transformation based methods transform the original intensity values of the pixels of the input image into more discriminant values. Bashir et al. (2010) proposed a intensity transformation based method for gait recognition. The method has used Shannon entropy of foreground probability in each pixel to compute the gait entropy image (GEnI). Li et al. (2016) presented another intensity transformation based method for gait recognition using GERFs. Attempt has also been made by combining intensity and spatial metric learning based methods. Makihara et al. (2017) proposed an approach of gait recognition by combining these two methods. The joint architecture has been optimized using linear support vector machine (SVM).

Deep learning based discriminative approaches have extracted deep invariant features from each walking pattern and provided better performance than traditional methods. Wu et al. (2017) proposed a deep learning based discriminative approach for gait recognition, where DCNNs were applied on a pair of probe and gait energy images. Shiraga, Makihara, Muramatsu, Echigo, and Yagi (2016) presented an eight-layered CNN architecture to extract the deep invariant features from gait energy images. Takemura et al. (2019) used a Siamese network to learn the deep invariant features corresponding to eachwalking pattern. Zhang, Huang et al. (2019), Zhang, Luo, Ma, Liu and Li (2019) and Zhang et al. (2019) presented a joint CNN based method for gait recognition. In another study (Li et al., 2019), Li et al. proposed a joint intensity transformer network to deal with COs and clothes in order to recognize the walking patterns. The method integrated intensity metric learning with deep network. Zhang, Huang et al. (2019), Zhang, Luo et al. (2019) and Zhang, Tran et al. (2019) combined the unique-gait and cross-gait networks for gait recognition and utilized the advantages of both the networks. In another recent study (Zhang, Huang et al., 2019; Zhang, Luo et al., 2019; Zhang, Tran et al., 2019), Zhang et al. proposed a gait recognition framework that disentangled pose and appearance features of a person during walking. In another study (Liu, Zhang, Ma, & Li, 2018), Liu et al. combined both deep learning based and traditional methods for gait recognition. In this study, spatial gait features from the GEIs and temporal gait features from the silhouette sequences were extracted initially. These features were then embedded by the null Foley–Sammon transform (NFST). Deng, Wang, Cheng, and Zeng (2017) presented a gait recognition system where spatial–temporal and kinematic features were fused. The system has relied on deterministic learning. Liao, Yu, An, and Huang (2020) presented a CNN based gait recognition method that has exploited the human 3D pose. Wang and Yan (2020) proposed one gait recognition method based on convolutional operations and LSTM. In a recent study (Gul, Malik, Khan, & Shafait, 2021), Gul et al. has proposed a spatio-temporal features based gait recognition system. The recognition phase has been carried out using DCNN. In another recent study (Sadeghzadehyazdi, Batabyal, & Acton, 2021), a CNN-LSTM based deep learning model has been proposed to study the spatiotemporal features from the skeleton data in order to recognize the gait anomaly. Li et al. (2022) proposed a human gait recognition method using spatio-temporal slice features. In a very recent study (Liu, You, He, Bi, & Wang, 2022), Liu et al. proposed a skeleton based gait recognition method using graph convolutional network.

However, discriminative approaches using both traditional and deep learning based methods cannot recognize a walking pattern correctly if the person (in walking) carries varying COs because these approaches learn discriminant features of each walking pattern with a few specific COs.

2.2. Studies using generative approaches

Studies on generative approaches are limited and they have focused on removing the COs from the original input images using the conventional detection and elimination strategies or deep generation networks like generative adversarial network (GAN).

Decann and Ross (2010) used the conventional detection and elimination strategy to remove the COs. The study presented a gait curve to grab the shape variability from silhouettes. Then the curves were restored without bags to enhance the gait recognition performance. Whytock et al. (2015) proposed a method that has combined the elimination method for single compact 2D GEIs and CO factor detection and obtained a improved representation of the input image without COs. The study proposed a bolt-on method using the conventional detection and elimination strategy to remove the COs.

A few studies have used deep generation networks as a generative approach to remove the COs from the original input images. The deep generation networks have been designed based on the GAN. Yu, Chen, Reyes, and Poh (2017) proposed the first deep generation network based model GaitGAN for gait recognition. GaitGAN removes the COs from the input images and generates an alternative representation of the input image with normal clothing. Li et al. (2020) presented a gait recognition method using alpha blending GANs. Here, after generating the gait template without COs, it has been blended with the original gait template to prevent the alteration of non-CO portions of the image. This blending has been done using an approximated alpha matte. In another study (Yu et al., 2019), Yu et al. proposed a GAN based method named GaitGANv2 for gait recognition. The method has extracted gait features invariant to view angle, clothing, presence of COs, and occlusion. He et al. (2019) presented MGANs to generate the alternative representation of the input images with different view angle. But, this method is not suitable for various types of COs.

However, the conventional approaches can deal only with the specific COs with which the system has been trained. It cannot deal with the situation if a person carries various types of COs like backpack, drawstring bag, satchel backpack, shoulder bag, waist bag, handheld bag, suitcase, briefcase, trolley bag, etc. Similarly, the GAN-based approaches successfully remove the COs from the input images, but these approaches may distort the non-CO parts during regeneration of the image, which may lead to unnecessary errors.

3. Dataset description

The present work has used four widely used public datasets OU-ISIR Large Population Gait database with real-life carried object (OU-LP-Bag) (Uddin et al., 2018), the OU-ISIR Gait database Treadmill dataset B (OUTD-B),[1] OU-ISIR Large Population Gait database with Age (OULP-Age),[2] and CASIA Gait database B (CASIA-B) (Yu, Tan, & Tan, 2006) for gait recognition. The details of these datasets are elaborated below.

3.1. OU-LP-Bag

OU-LP-Bag is the largest gait database available currently with various types of COs. It contains the walking activity video frames of 62,528 different persons with seven different situations, i.e., no COs (NoCO), side bottom COs (SbCO), side middle COs (SmCO), front COs (FrCO), back COs (BaCO), multiple objects carried in different locations (MuCO), and the carrying locations of objects change during walking (CpCO). Ages of the subjects are ranging from 2 to 95 years. Three different walking activity videos are there for each subject; two of them are without COs and one is with CO. A few GEI examples from OU-LP-Bag dataset is shown in Fig. 2. All the subjects have been included in the training and testing sets. Varying numbers of frames are associated with each walking activity video. In the present investigation, 60 number of frames have been considered from each walking activity video. These frames have been equally divided into training and testing sets.

3.2. OUTD-B

OUTD-B dataset contains video frames during walking on the treadmill of 68 different subjects with up to 32 clothing variations and without COs. All the subjects have been included in the training and testing sets. Varying numbers of frames are associated with each walking activity video. In the present investigation, 300 different frames have been considered from each walking activity video. These frames have been equally divided into training and testing sets. A few GEI frames from OUTD-B dataset is shown in Fig. 3.

[1] http://www.am.sanken.osaka-u.ac.jp/BiometricDB/GaitTM.html.

[2] http://www.am.sanken.osaka-u.ac.jp/BiometricDB/GaitLPAge.html.

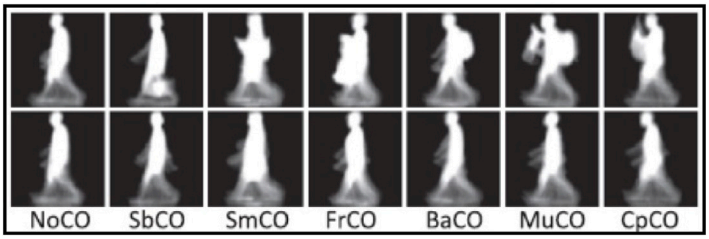

Fig. 2. Examples of GEIs with seven different situations from OU-LP-Bag dataset. Each column shows the same subject. The first row shows different subjects with a CO and the second row shows the subjects without a CO.

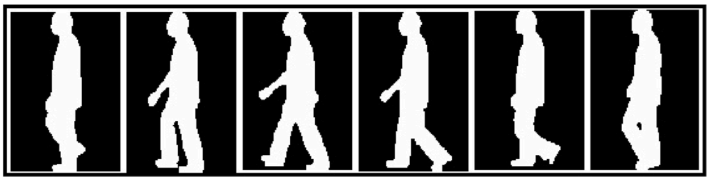

Fig. 3. A few GEI frames from OUTD-B dataset.

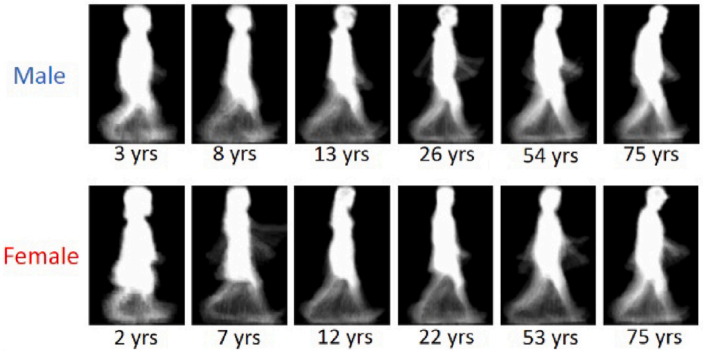

Fig. 4. A few GEI frames from OULP-Age dataset. The first row shows the frames of different male subjects with varying ages and the second row shows different female subjects with varying ages.

3.3. OULP-Age

This dataset contains video frames of walking activity of 63,846 subjects with ages ranging from 2 to 90 years old. All the subjects have been included in the training and testing sets. Varying numbers of frames are associated with each walking activity video. In the present investigation, 300 different frames have been considered from each walking activity video. These frames have been equally divided into training and testing sets. A few GEI frames of different subjects with varying ages from OULP-Age dataset are shown in Fig. 4.

3.4. CASIA-B

CASIA-B dataset contains walking activity video frames of 124 subjects with 110 sequences per subject. These sequences have been collected in 11 different view angles between 0° and 180°. There are 10 walking sequences per view angle. Out of these 10 sequences, 6 are without carrying any object, 2 are with carrying bags, and other 2 are with wearing coats. All the subjects have been included in the training and testing sets. Varying numbers of frames are associated with each walking activity sequence. These frames have been equally divided into training and testing sets. Frames of walking activity without carried object in 11 different view angles are shown in Fig. 5.

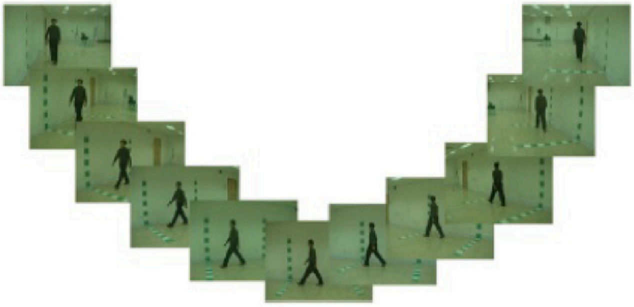

Fig. 5. Frames of walking activity without carried object in 11 different view angles from CASIA-B dataset.

4. Proposed method

In the proposed method, the person present in the frames (GEIs) of any walking activity video has been first detected through a modified version of the traditional Faster R-CNN. The discriminant features have then been extracted using deep convolutional layers from the walking pattern of the detected person, generated comprising all the frames (GEIs). The dimension of each GEI has been fixed to 128 × 128. The generated feature vector of the walking pattern has then been recognized using RNN classifier, a deep learning network. The overall block diagram of the proposed method is shown in Fig. 6.

4.1. Pedestrian detection in the frames (GEIs)

This work proposes a novel method of detecting the pedestrian present in the GEIs of any walking activity video irrespective of the

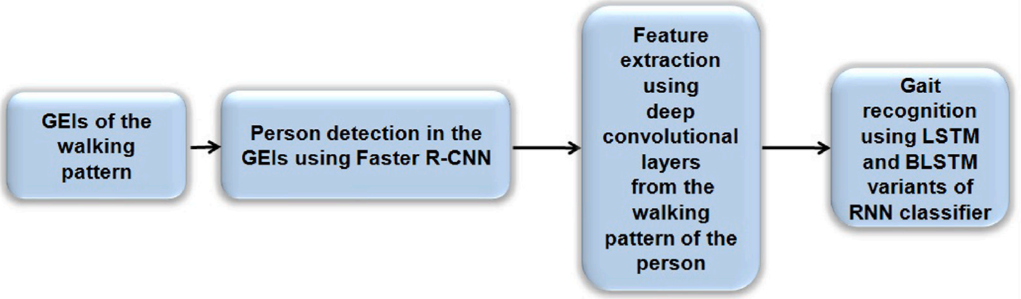
GEIs of the walking pattern
Person detection in the GEIs using Faster R-CNN
Feature extraction using deep convolutional layers from the walking pattern of the person
Gait recognition using LSTM and BLSTM variants of RNN classifier

Fig. 6. Overall structure of the proposed method.

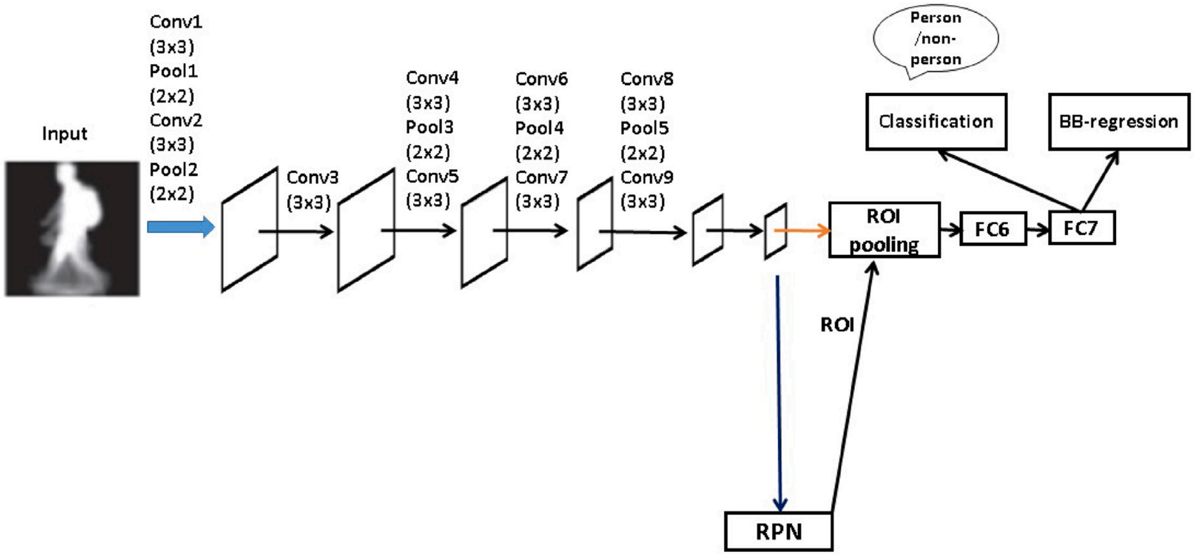
Input
Conv1 (3x3) Pool1 (2x2) Conv2 (3x3) Pool2 (2x2)
Conv3 (3x3)
Conv4 (3x3) Pool3 (2x2) Conv5 (3x3)
Conv6 (3x3) Pool4 (2x2) Conv7 (3x3)
Conv8 (3x3) Pool5 (2x2) Conv9 (3x3)
ROI pooling
FC6
FC7
Classification
BB-regression
Person /non-person
ROI
RPN

Fig. 7. The proposed modified version of traditional Faster R-CNN architecture.

Table 1
Statistics of various anchors used in the traditional Faster R-CNN.

Scale	Size(Height × Width)		
1:2	95 × 183	191 × 367	383 × 735
1:1	127 × 127	255 × 255	511 × 511
2:1	175 × 87	351 × 175	703 × 351

Table 2
Statistics of different anchors used in the proposed modified version of conventional Faster R-CNN.

Scale	Size(Height × Width)		
3:1	60 × 20	75 × 25	90 × 30
4:1	80 × 20	100 × 25	120 × 30
4.5:1	90 × 20	113 × 25	135 × 30

person is carrying any CO or not, using a modified version of Faster R-CNN architecture. The traditional Faster R-CNN i.e., original VGG16 model uses five different convolutional layers, whereas the proposed modified form has used nine different convolutional layers. Apart from this, a single RPN in the traditional Faster R-CNN produces several region proposals utilizing the feature maps generated by the last convolutional layer and nine different anchor boxes with varying aspect ratios are used to generate the proposal boxes. But, the standard aspect ratios of the anchor boxes have been modified in the proposed modified version of the conventional Faster R-CNN because the anchor boxes with standard aspect ratios could not deliver satisfactory performance in this work in detecting the pedestrians of varying heights and widths. The original aspect ratios used in the traditional Faster R-CNN and the aspect ratios proposed in this work of different anchor boxes are shown in Tables 1 and 2 respectively. The proposed modified version of traditional Faster R-CNN architecture is shown in Fig. 7.

4.1.1. Region proposal network

The RPN uses a small network where a single window slides over the feature matrix obtained from the last convolutional layer to generate the region proposals. Each sliding window has been mapped to 256-D Zeiler and Fergus network (Zeiler & Fergus, 2014) to obtain spatial features for every location. The RPN predicts multiple region proposals for each location of the feature matrix simultaneously. As many anchors are there in one sliding window, that many region proposals are generated for each location of the feature matrix. The centre point of the anchor box coincides with the centre point of each sliding window. Different aspect ratios and scales are associated with these anchors. In the present work, three different scales and three different aspect ratios, yielding a total of nine anchor boxes for each location have been used. The statistics of different anchor boxes used in the present investigation are presented in Table 2. The positive labels have been assigned to the anchors satisfying the following condition:

Condition: If the intersection-over-union (IoU) overlap value between the anchor box and ground truth (GT) box is greater than 0.8, then assign positive label to the anchor box.

All the positive labelled anchor boxes generated by RPN have been fed as input to the ROI pooling layer.

4.1.2. Region of interest pooling

In the present work, the RPN has produced several region proposals of varying sizes. Each proposal is represented using a five tuple (i, x, y, w, h), where i indicates the index of the proposal, and (x, y, w, h) are the coordinates of the rectangular detecting box (proposal). The ROI pooling layer has projected these proposals onto the feature matrix obtained from the last convolutional layer. The inputs to the ROI pooling layer are the feature matrix and the positive labelled proposals generated by the RPN. In the present investigation, 100 ROI proposals with higher IoU value out of the total positive labelled proposals generated by RPN have been selected by the ROI pooling layer. The optimal value of number of ROI proposals has been found using *Bayesian optimization technique*. The ROI pooling layer has also performed the max-pooling operation on the feature matrix corresponding to each proposal generated by the RPN.

4.1.3. Detection of pedestrian

The proposed architecture in the present investigation has performed two different tasks—predicting whether the pedestrian is present or not in the frame and a bounding box around the detected person. The ROI obtained from the ROI pooling layer is sent to the fully connected (FC) layers as input one at a time. The softmax function has been used as an activation function in the output layer to predict the existence of the person. Apart from this, a regressor has been used to generate the rectangular box coordinates around the detected person. In the proposed architecture, the classification loss (Ren et al., 2015) and regression loss (Ren et al., 2015) have been computed at the output layer of the detection network.

4.1.4. Training scheme

The proposed architecture has been trained by an end-to-end method. Each frame (GEI) of any training video sample has been fed to the proposed architecture of faster R-CNN. The GT of each frame of the training sample has been developed manually. The learning rate and the weight decay (gamma) has been set to 0.025 and 0.05 respectively to train the proposed architecture. The loss calculated at the output layer of the detection network has been back propagated to the architecture for fine tuning. The classification loss and regression loss have been calculated during the training of the proposed architecture. The classification loss measures the classification correctness of each detecting box. The regression loss measures the closeness of the coordinates of the detecting box around the detected person.

4.2. Extracting the features of walking pattern of detected pedestrian

Once the pedestrian detection process in each frame gets over, various feature values have been extracted from the walking pattern of the detected pedestrian using deep convolutional layers. The walking pattern of the pedestrian has been obtained through numerous frames, in fact all the frames of any walking activity video considered in this work for training or testing purpose. All of these frames generating the walking pattern of the pedestrian have been fed as input to the first convolution layer. Six different convolution layers (conv1, conv2, conv3, conv4, conv5, and conv6) have been used in the present system to extract the features. The first three convolution layers (conv1, conv2, and conv3) have been applied consecutively. One maxpooling layer (pool1) has been incorporated after these three convolution layers. Two more consecutive convolution layers (conv4 and conv5) have been incorporated after pool1. Another maxpooling layer (pool2) has been incorporated after conv5. Finally, another convolution layer (conv6) has been used followed by the final maxpooling layer (pool3). Convolution layers extract features using filters and the pooling layers reduce the dimension of those feature maps to make it computationally efficient for further layers. Convolutional layers take the inner product of the linear filter and underlying receptive field, followed by a nonlinear activation function at every local part of the input. This operation can be expressed as

$$y_i^l = f(\sum_i^{n-1} W_{pq} * x_i^{l-1} + b_i), \tag{1}$$

where y_i^l is the ith output of the lth convolution layer, $f(.)$ is an activation such as the rectified linear unit, W_{pq} is the trainable filters, x_i^{l-1} is the last feature maps or input data, b_i are the biases, and the symbol $*$ is a discrete convolution operator. The resulting outputs are called feature maps. The pooling layers use the maximum (or average) value of the receptive field at every local part of the feature maps. The use of convolutional layers reduces the computational cost of extracting the features. Also, the convolutional layers can learn the discriminating features of the walking patterns from the input image, making classification simple for further layers.

4.3. Recognition of the walking pattern

In the present work, the walking patterns have been recognized i.e., the feature vectors generated after pool3 have been classified using LSTM and BLSTM variants of RNN classifier. The internal states of RNN can remember the inputs of several past timestamps due to the existence of recurrently-connected nodes in the hidden layers. RNNs are models that consist of standard recurrent cells, shown in Fig. 8. The typical feature of the RNN cell is a cyclic (or loop) connection, which enables the model to update the current state based on past states and current input data. Formally, the standard recurrent cell is defined as follows:

$$h_j = \phi(W_h h_{j-1} + W_z z_j + b) \tag{2}$$

$$o_j = h_j \tag{3}$$

where $z_j = (x, y, t)_j$ denotes the jth vector of the input signal z $= (x, y, t)_{j=1,\ldots,|z|}$ at timestep j, h_j is the hidden state of the cell, and o_j denotes the cell output, respectively; W_h and W_z are the weight matrices; b is the bias of the neurons; and ϕ is an activation function. Standard recurrent cells have achieved success in many sequence learning problems such as handwriting recognition (Ghosh et al., 2019), action recognition (Du, Wang, & Qiao, 2017), or image captioning (Mao et al., 2014). However, the standard recurrent cells are not capable of handling long-term dependencies. To solve this issue, the LSTM cells were developed (Ghosh et al., 2019; Graves et al., 2009). LSTM cells improve the capacity of the standard recurrent cell by introducing different gates, which are briefly described below.

The LSTM cell is defined as follows:

$$G_{ip} = \sigma(W_{ud}[h_{j-1}, z_j] + b_{ip}) \tag{4}$$

$$G_{fg} = \sigma(W_{fg}[h_{j-1}, z_j] + b_{fg}) \tag{5}$$

$$G_{op} = \sigma(W_{op}[h_{j-1}, z_j] + b_{op}) \tag{6}$$

$$c_j = G_{ud} \circ \tilde{c}_j + G_{fg} \circ c_{j-1} \tag{7}$$

$$\tilde{c}_j = \phi(W_c[h_{j-1}, z_j] + b_c) \tag{8}$$

$$h_j = G_{op} \circ \phi(c_j) \tag{9}$$

where c_j is an additional hidden state, W_* are weight matrices, b_* are biases, G_* denote cell gates (ip: input, fg: forget, op: output, ud: update), and ϕ and σ are activation functions (hyperbolic tangent and sigmoid, respectively). The operator $\circ$ denotes the Hadamard (element-wise) product. Fig. 9 shows the organization of one LSTM cell.

It may be noted that the LSTM has two kinds of hidden states: a "slow" state c_j that keeps long-term memory, and a "fast" state h_j that

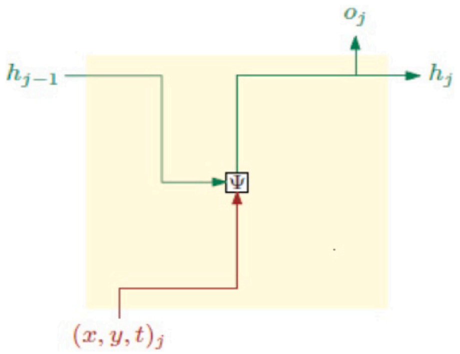

Fig. 8. A simple RNN cell.

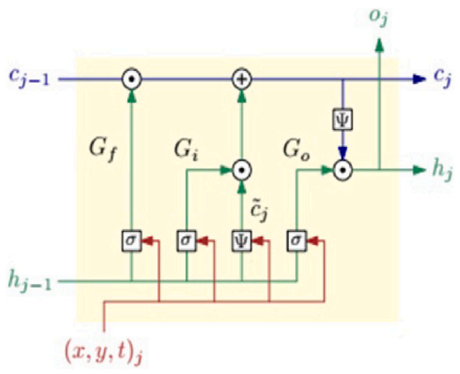

Fig. 9. A LSTM cell.

makes decisions over short periods of time. The forget gate decides which information will be kept in the cell state and which information will be thrown away from the cell state. Apart from hyperbolic tangent and sigmoid activation functions, there are various other non-linear activation functions have been promoted in the research literature. In the present work, as RNN receives the input from the convolutional layers, so it is already receiving complex features extracted by the convolutional layers.

Accessing of both future and past contexts are required in several tasks to predict the correct class label of the sample. For example, if the class label of any walking pattern is predicted by accessing both future and past GEIs with respect to a set of GEIs, then the prediction will be more accurate. Bidirectional RNNs (BRNNs) (Schuster & Paliwal, 1997) are capable to access context in forward as well as backward directions along the sequence. BRNNs have two separate hidden layers—one processes the sequence in forward direction and the second one processes it in backward direction. The same output layer remains connected with both of these hidden layers and it enables the accessing of both past and future contexts of the sequence. Combination of BRNNs and LSTM gives rise to BLSTM variant of RNN.

Both LSTM and BLSTM variants of RNN have been implemented using the *theano* toolkit.[3] Dataset wise training times are not same using this toolkit due to the existence of varying numbers of training samples in different datasets. It has taken 5 to 7 min to train one sample in LSTM model, whereas the training time has varied between 8 and 10 min per sample in BLSTM model. To train the RNN classifier, each feature vector generated by the deep convolutional layers has been labelled by a class label. The feature vectors generated from different training samples of a single walking pattern have been labelled by the same class label. The number of classes in the training set has been set as equal to the number of subjects in each dataset. All the subjects present in each dataset have been considered in the training set. This training set has been used to train both LSTM and BLSTM models of RNN. During testing, the feature vectors of different testing samples have been fed to both LSTM and BLSTM models of RNN to recognize the walking patterns corresponding to these test feature vectors.

5. Experimental results and analysis

The experiments on person detection in the GEIs and gait recognition have been performed on four widely used public datasets discussed in Section 3. All the subjects have been included in training and testing sets in all the four datasets.

5.1. Parameters of convolutional layers

For each convolution operation, a 2D filter of dimension 3 × 3 has been used. 64 filters have been used for the first three convolution layers (conv1, conv2, and conv3), whereas 128 filters have been used for the next two convolution layers (conv4 and conv5). For the final convolution layer (conv6), 256 filters have been used. Pooling operation has been applied on 2 × 2 dimensional sub-matrix of the feature matrix with a stride of 2. The final feature matrix obtained after the last maxpooling layer (pool3) has a dimension of $16 \times 16 \times (n)$, where n denotes the number of frames considered for training/testing from a single video of walking activity. This n varies from dataset to dataset. The final feature matrix obtained after performing pool3 has been flattened and fed as input to the RNN to classify the feature vector.

5.2. Hyperparameters of RNN

In this work, LSTM model contains 1 (optimal value) forward hidden layer. On the other hand, BLSTM model contains 2 (optimal value) separate hidden layers. Among these two hidden layers, one processes the input sequence in forward direction and the other processes it in backward direction. These optimal values of number of hidden layers have been obtained using *Bayesian optimization* technique. The input layer of RNN has accepted a $16 \times 16 \times (n)$ (the significance of n is mentioned in Section 5.2) dimensional feature vector for LSTM and BLSTM variants. Each LSTM hidden layer contains 55 recurrently connected memory blocks, whereas each BLSTM hidden layer also contains 55 recurrently connected memory blocks. The experiments have been carried out with varying numbers of memory blocks in each LSTM and BLSTM hidden layer. But, the *Bayesian optimization* technique has provided the optimal value of 55 for the number of memory blocks in each LSTM and BLSTM hidden layer. The optimal values of the hyperparameters epoch and batch size are 50 and 8 respectively, obtained using *Bayesian optimization* technique. Hyperbolic tangent activation function has been used in each memory block as input and output activation functions. The gates have been activated using the sigmoid activation function. The number of neurons in the output layer of RNN depends on the number of subjects present in a particular dataset. For experimentations with the OUTD-B dataset, 68 neurons have been incorporated in the output layer because this dataset contains the walking activity videos of 68 different subjects. Similarly, for experimentations with CASIA-B dataset, 124 neurons have been incorporated in the output layer because 124 different subjects are present in this dataset. The softmax activation function has been used to activate the output layer neurons. The network has been optimized using the Adam optimizer algorithm (Kingma & Ba, 2014) and categorical cross-entropy loss function with the learning rate of 0.0001. The optimal set of values of various hyperparameters of RNN obtained in the present work is presented in Table 3.

[3] http://deeplearning.net/software/theano/.

Table 3
The optimal set of values of various hyperparameters of RNN obtained in the present work.

Hyperparametrs	Values
LSTM hidden layers	1
BLSTM hidden layers	2
Memory blocks in each LSTM and BLSTM layer	55
Batch size	8
Epoch	50
Learning rate	0.0001
Activation function in output layer	Softmax activation

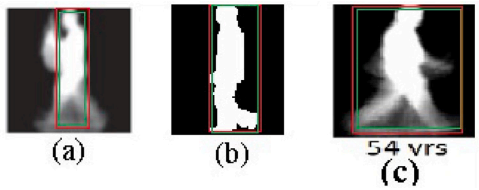

Fig. 10. Correct detection of persons using the proposed detection method on a few test GEIs from (a) OU-LP-Bag, (b) OUTD-B, and (c) OULP-Age datasets. The red bounding box represents the ground-truth, whereas the green one is the detecting box.

5.3. Pedestrian detection results

The performance of detecting the pedestrian in each GEI using the modified form of the conventional Faster R-CNN has been evaluated using *accuracy*, *precision*, *recall*, and *F1-Score* metrics. If the IoU value between the detecting and GT boxes is greater than 0.8, then the detection of the person is considered as *true positive*, whereas *false positive* has been considered when more than one detecting boxes are generated on the person in the GEI. *False negative* has been considered when no detecting box has been generated despite the presence of the person in the GEI. It happens only when the IoU value is below 0.8. Table 4 presents the pedestrian detection results using the proposed Faster R-CNN architecture in terms of *accuracy*, *precision*, *recall*, and *F1-Score*. Table 5 presents the pedestrian detection results using YOLOv3. The experiment has also been carried out using Cascade R-CNN (Cai & Vasconcelos, 2018), an extended version of Faster R-CNN. The performance of Faster R-CNN generally degrades if the IoU threshold is set to a very high value. Cascade R-CNN solves this problem. But Cascade R-CNN has not shown better performance in comparison to Faster R-CNN in the present work for the IoU threshold value of 0.8. Cascade R-CNN has shown better performance than Faster R-CNN when the IoU threshold value has been set to 0.9, but that performance is still inferior than the best performance shown by Faster R-CNN for the IoU threshold value of 0.8. The pedestrian detection results in terms of *accuracy* using Cascade R-CNN as well as Faster R-CNN for IoU threshold values of both 0.8 and 0.9 are presented in Table 6. The results from Tables 4, 5, and 6 demonstrate that the pedestrian detection performance of the proposed Faster R-CNN architecture is much more superior in comparison to YOLOv3 architecture as well as Cascade R-CNN. Table 7 presents the computation speed of person detection in Frames per second (fps). The present work has been implemented on a single Titan Xp GPU. The optimal set of values of various hyper parameters used in the Faster R-CNN architecture in the present work are shown in Table 8. This optimal set has been found using *Bayesian optimization* technique. Table 9 presents the pedestrian detection *accuracies* for various combinations of values of hyperparameters of Faster R-CNN. Fig. 10 illustrates the correct detection of person in a few test GEIs from three different datasets. Detected persons are confined within green coloured boxes in this figure.

Table 4
Pedestrian detection results using the proposed Faster R-CNN architecture in terms of accuracy, precision, recall, and F1-Score.

Dataset	Accuracy	Precision	Recall	F1-Score
OU-LP-Bag	98.74%	98.52%	98.58%	98.54%
OUTD-B	99.67%	99.46%	99.53%	99.50%
OULP-Age	99.69%	99.51%	99.55%	99.52%
CASIA-B	99.41%	99.26%	99.32%	99.28%

Table 5
Pedestrian detection results using YOLOv3.

Dataset	Accuracy	Precision	Recall	F1-Score
OU-LP-Bag	93.68%	93.45%	93.51%	93.47%
OUTD-B	94.54%	94.35%	94.39%	94.36%
OULP-Age	94.57%	94.38%	94.42%	94.39%
CASIA-B	94.45%	94.31%	94.37%	94.33%

Table 6
Pedestrian detection accuracies using Cascade R-CNN and Faster R-CNN.

Dataset	IoU threshold value	Accuracy	
		Cascade R-CNN	Faster R-CNN
OU-LP-Bag	0.8	96.23%	98.74%
	0.9	97.54%	97.18%
OUTD-B	0.8	96.68%	99.67%
	0.9	98.12%	97.84%
OULP-Age	0.8	96.72%	99.69%
	0.9	98.18%	97.89%
CASIA-B	0.8	96.54%	99.41%
	0.9	98.06%	97.78%

Table 7
Processing speed for person detection using Faster R-CNN.

Dataset	Processing speed (in fps)
OU-LP-Bag	54
OUTD-B	56
OULP-Age	56
CASIA-B	58

Table 8
The list of hyper parameters used in the Faster R-CNN architecture and their optimal values.

Hyper parameter	Optimal value
Batch size	512
Overlap threshold for ROI	0.8
Number of ROIs	100
Learning rate	0.025
Weight decay for regularization	0.05

5.4. Gait recognition results

The experiments have been performed with various combinations of *blocks*, *epoch*, and *batch size* (Ghosh et al., 2019) values in both the variants of RNN classifier. The detailed analyses of gait recognition *accuracy* of the present system against varying combinations of *blocks*, *epoch*, and *batch size* values are presented in Table 10. The results in Table 10 show that the best recognition accuracy has been obtained for the combination "*blocks* = 55, *epoch* = 50, and *batch size* = 8" in all of the four datasets. Table 10 also shows that BLSTM model provides better recognition accuracy over LSTM model. Fig. 11 presents various top choices of gait recognition accuracy using both the models of RNN. The gait recognition performance has also been measured using *precision*, *recall*, and *Receiver Operator Characteristic* (*ROC*) *curve* metrics. The *precision*, *recall*, and *ROC curve* of gait recognition of the present work using BLSTM model are presented in Figs. 12, 13, and 14 respectively.

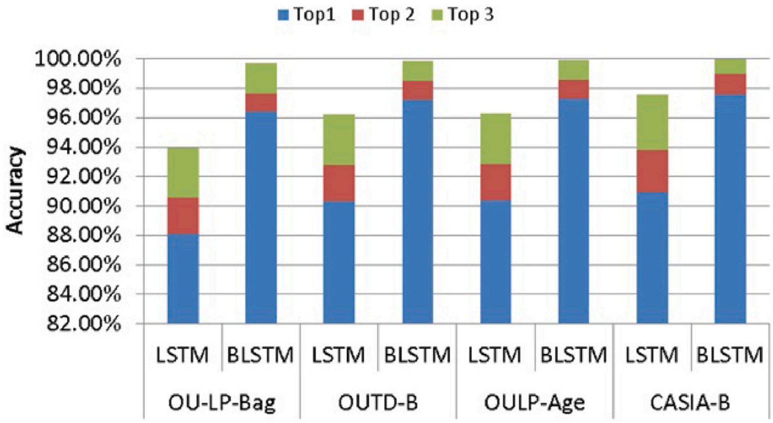

Fig. 11. Gait recognition *accuracy* of various top choices of the proposed system.

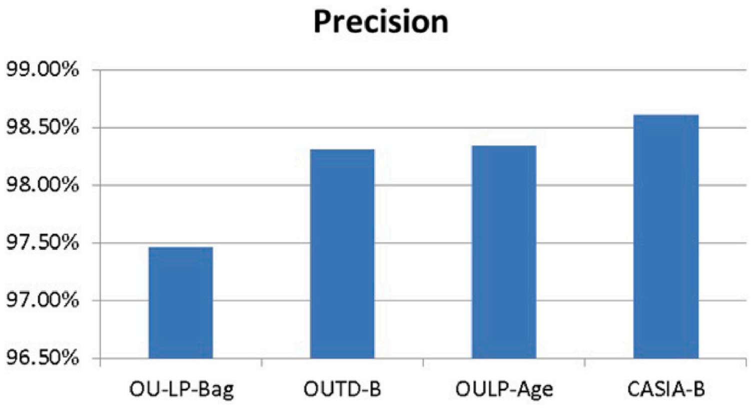

Fig. 12. *Precision* of gait recognition of the proposed system using BLSTM.

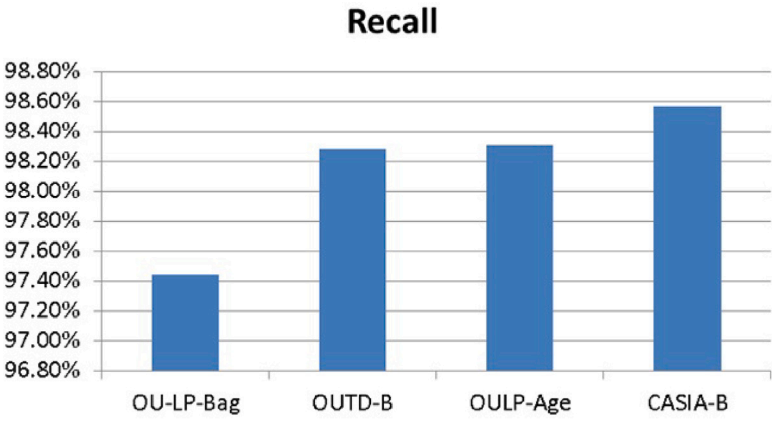

Fig. 13. *Recall* of gait recognition of the proposed system using BLSTM.

5.5. Computational complexity of gait recognition

The computational complexity of gait recognition in the present work has varied from dataset to dataset. The computational complexity of gait recognition has increased for two big datasets OU-LP-Bag and OULP-Age due to the existence of large number of classes in these two datasets, whereas the computational complexity has remained comparatively lower for other two datasets. The present work has been implemented on a single Titan Xp GPU. Fig. 15 presents the computational time of gait recognition per test sample using the BLSTM variant of RNN classifier for all the four datasets.

5.6. Comparison with state-of-the-art results

Table 11 compares the performance of the proposed gait recognition system with some of the existing significant gait recognition systems on OU-LP-Bag, OUTD-B, and CASIA-B datasets. The recognition results are presented in terms of accuracy.

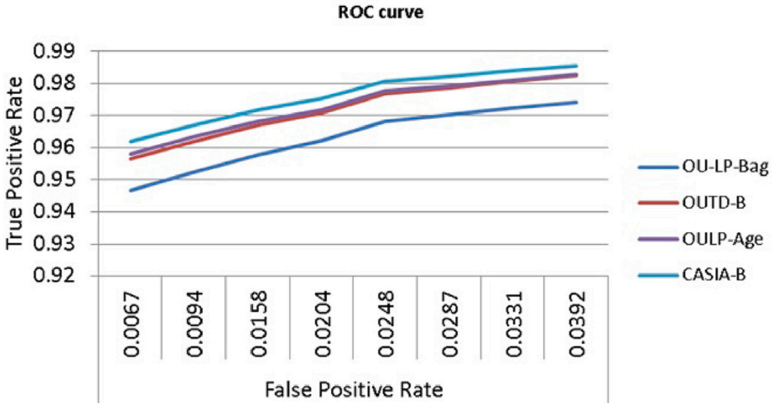

Fig. 14. *ROC curve* of gait recognition of the proposed system using BLSTM.

Table 9
Pedestrian detection *accuracies* for various combinations of values of hyperparameters of Faster R-CNN.

Dataset	Batch size	Overlap threshold for ROI	Number of ROIs	Accuracy
OU-LP-Bag	768	0.7	120	82.34%
	512	0.7	120	86.78%
	768	0.8	100	95.58%
	512	0.8	100	**98.74%**
OUTD-B	768	0.7	120	82.88%
	512	0.7	120	87.36%
	768	0.8	100	96.28%
	512	0.8	100	**99.67%**
OULP-Age	768	0.7	120	82.89%
	512	0.7	120	87.37%
	768	0.8	100	96.29%
	512	0.8	100	**99.69%**
CASIA-B	768	0.7	120	82.48%
	512	0.7	120	87.02%
	768	0.8	100	95.98%
	512	0.8	100	**99.41%**

Table 10
Gait recognition *accuracy* of the present system.

Dataset	Blocks	Epochs	Batch size	Accuracy	
				LSTM	BLSTM
OU-LP-Bag	25	50	20	43.68%	49.86%
	25	50	8	52.54%	60.31%
	55	50	20	79.58%	87.62%
	55	50	8	89.12%	**97.42%**
OUTD-B	25	50	20	44.68%	50.72%
	25	50	8	54.42%	62.74%
	55	50	20	81.38%	89.46%
	55	50	8	91.34%	**98.24%**
OULP-Age	25	50	20	44.71%	50.74%
	25	50	8	54.45%	62.76%
	55	50	20	81.42%	89.49%
	55	50	8	91.37%	**98.28%**
CASIA-B	25	50	20	45.23%	51.62%
	25	50	8	54.89%	63.12%
	55	50	20	82.16%	89.88%
	55	50	8	91.92%	**98.54%**

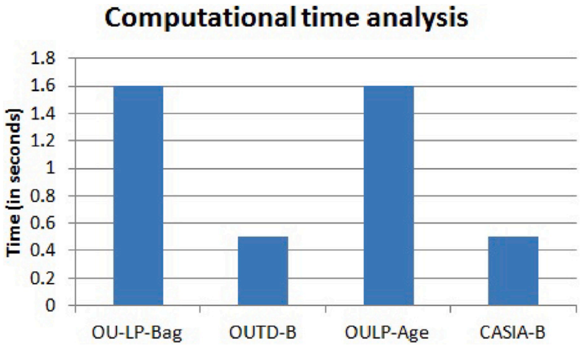

Fig. 15. Computational time of gait recognition per test sample using the BLSTM variant of RNN classifier.

5.7. Strengths of the proposed system

The major strengths of the proposed system are listed below.

- The existing gait recognition systems based on discriminative and generative approaches have various limitations in recognizing different walking patterns if persons carry various COs. The discriminative approaches cannot recognize a walking pattern correctly if the concerned person carries various types of COs because these approaches learn discriminant features of each walking pattern with a few specific COs. Similarly, the existing generative approaches have various limitations like the conventional approach can deal only with the particular type of COs. It cannot deal with the situation if a person carries various types of COs like backpack, drawstring bag, satchel backpack, shoulder bag, waist bag, handheld bag, suitcase, briefcase, trolley bag, etc. Similarly, the GAN-based approaches successfully remove the COs from the input images, but these approaches regenerate other non-CO parts as well, which may lead to unnecessary errors. The present investigation overcomes these drawbacks of the existing systems by detecting the person only from the input image and not the CO using Faster R-CNN architecture. Faster R-CNN is more capable to detect various objects accurately in comparison to other CNN based architectures due to the generation of various region proposals using the region proposal algorithm.
- The proposed system has used LSTM and BLSTM models of RNN classifier to recognize the walking patterns. RNN can remember a long sequence of symbols for a long duration through these two versions. Apart from this, BLSTM remembers any sequence in both the directions. Walking activity video of any individual is a collection of several frames in a sequential order. These constituent frames of the video generate the entire walking pattern. By exploiting the sequence memorizing capability of these two variants of RNN, the proposed system recognizes any walking pattern by remembering the sequential order of the concerned frames.
- The high accuracy of the proposed system makes it suitable for deployment in real life applications.

Table 11
Comparison of the performance of the proposed gait recognition system with some existing significant gait recognition systems.

Dataset	Study	Method	Accuracy
OU-LP-Bag	Li et al. (2020)	Alpha blending generative adversarial networks	78.11%
	Li et al. (2019)	Joint intensity transformer network	74.44%
	Takemura et al. (2019)	CNN based cross-view gait recognition	73.14%
	Wu et al. (2017)	Similarity learning using deep CNNs	74.39%
	Chopra, Hadsell, and LeCun (2005)	Similarity metric based learning	49.80%
	Gul et al. (2021)	Spatio-temporal features and DCNN	95.16%
	Proposed system	Faster R-CNN and RNN based approach	**97.42%**
OUTD-B	Li et al. (2019)	Joint intensity transformer network	89.6%
	Takemura et al. (2019)	CNN based cross-view gait recognition	89.1%
	Guan et al. (2015)	Classifier ensemble method	90.7%
	Wu et al. (2017)	Similarity learning using deep CNNs	87.3%
	Lee, Tan, and Tan (2013)	Fourier descriptor	84%
	Lee, Tan, and Tan (2014)	Probabilistic gait representation	84%
	Felez and Xiang (2014)	Probabilistic gait representation	98%
	Proposed system	Faster R-CNN and RNN based approach	**98.24%**
CASIA-B	Wu et al. (2017)	Similarity learning using deep CNNs	91.1%
	Zhang, Huang et al. (2019), Zhang, Luo et al. (2019) and Zhang, Tran et al. (2019)	Joint Unique-gait and Cross-gait Network	95.9%
	Zhang, Huang et al. (2019), Zhang, Luo et al. (2019) and Zhang, Tran et al. (2019)	Autoencoder based CNN	91.5%
	Deng et al. (2017)	Fusion of spatial–temporal and kinematic features	97%
	Liu, Ye, Li, Zhang, and Lin (2016)	Memory based gait recognition	83.87%
	Kumar et al. (2019)	Decision fusion based approach	89.71%
	Gul et al. (2021)	Spatio-temporal features and DCNN	98.14%
	Proposed system	Faster R-CNN and RNN based approach	**98.54%**

6. Conclusion and future scope

The present work proposes a machine vision based method of gait recognition both with and without COs using Faster R-CNN and RNN. Due to the capability of Faster R-CNN in detecting the real-life objects more accurately in comparison to other object detection architectures, the pedestrian detection module in the present work has employed Faster R-CNN to detect the pedestrian only and not the CO in the GEIs. The reason behind detecting only the pedestrian when pedestrian is carrying any CO is to generate the similar feature vectors from the walking pattern of the pedestrian when the same pedestrian is walking without CO and with CO. The performance of the present system has been evaluated on four widely used public gait datasets with one of those containing COs and the results demonstrate that the proposed gait recognition method outperforms the state-of-the-art methods. The present system also overcomes various drawbacks of the existing systems as discussed in Section 5.7. The performance of the present system tells that the proposed gait recognition system could be employed for human identification in real world scenarios. The proposed system shall provide fresh insight to the researchers in developing gait recognition systems using computer vision based techniques.

However, the present system has misclassified several walking patterns when the pedestrian is carrying an object (such as sidebag hanging from the shoulder) that is in the same line with the leg of the pedestrian. In this situation, the CO could not be removed properly by the proposed pedestrian detection method and as a consequence the feature vector generated from the walking pattern of the pedestrian became dissimilar with the feature vector generated from the walking pattern of the same pedestrian not carrying any object. In future, the attempt will be made to overcome this drawback. It has been planned to carry out the research work in future in this domain by employing other advanced deep learning techniques. It has also been planned to develop a gait recognition system considering the clothing variations of persons while walking. The attempt will also be made to explore the possibility of utilizing the proposed model for person detection as well as gait recognition if more than two distinct objects exist in the input frames.

CRediT authorship contribution statement

Rajib Ghosh: Conceptualization, Data curation, Formal analysis, Investigation, Methodology, Resources, Software, Validation, Visualization, Writing – original draft, Writing review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

Bashir, K., Xiang, T., & Gong, S. (2010). Gait recognition without subject cooperation. *Pattern Recognition Letters*, *31*(13), 2052–2060.

Ben, X., Zhang, P., Lai, Z., Yan, R., Zhai, X., & Meng, W. (2019). A general tensor representation framework for cross-view gait recognition. *Pattern Recognition*, *90*, 87–98.

Bouchrika, I., Goffredo, M., Carter, J., & Nixon, M. (2011). On using gait in forensic biometrics. *Journal of Forensic Science*, *56*(4), 882–889.

Cai, Z., & Vasconcelos, N. (2018). Cascade R-CNN: Delving into high quality object detection. In *Computer vision and pattern recognition, IEEE computer society conference* (pp. 6154–6162). IEEE.

Chopra, S., Hadsell, R., & LeCun, Y. (2005). Learning a similarity metric discriminatively, with application to face verification. In *Computer vision and pattern recognition, IEEE computer society conference* (pp. 539–546). IEEE.

Decann, B., & Ross, A. (2010). Gait curves for human recognition, backpack detection, and silhouette correction in a nighttime environment. In *Biometric technology for human identification, SPIE conference* (pp. 1–13). SPIE.

Deng, M., Wang, C., Cheng, F., & Zeng, W. (2017). Fusion of spatial–temporal and kinematic features for gait recognition with deterministic learning. *Pattern Recognition*, *67*, 186–200.

Du, W., Wang, Y., & Qiao, Y. (2017). Recurrent spatial–temporal attention network for action recognition in videos. *IEEE Transactions on Image Processing*, *27*(3), 1347–1360.

Felez, R. M., & Xiang, T. (2014). Uncooperative gait recognition by learning to rank. *Pattern Recognition*, *47*, 3793–3806.

Ghosh, R., Vamshi, C., & Kumar, P. (2019). RNN Based online handwritten word recognition in Devanagari and Bengali scripts using horizontal zoning. *Pattern Recognition*, *92*, 203–218.

Graves, A., Liwicki, M., Fernandez, S., Bertolami, R., Bunke, H., & Schmidhuber, J. (2009). A novel connectionist system for unconstrained handwriting recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *31*(5), 855–868.

Guan, Y., Li, C. T., & Roli, F. (2015). On reducing the effect of covariate factors in gait recognition: a classifier ensemble method. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *37*(7), 1521–1528.

Gul, S., Malik, M. I., Khan, G. M., & Shafait, F. (2021). Multi-view gait recognition system using spatio-temporal features and deep learning. *Expert Systems with Applications*, *179*, Article 115057.

Han, J., & Bhanu, B. (2006). Individual recognition using gait energy image. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *28*(2), 316–322.

He, Y., Zhang, J., Shan, H., & Wang, L. (2019). Multi-task GANs for view-specific feature learning in gait recognition. *IEEE Transactions on Information Forensics and Security*, *14*(1), 102–113.

Iwama, H., Muramatsu, D., Makihara, Y., & Yagi, Y. (2013). Gait verification system for criminal investigation. *IPSJ Transactions on Computer Vision and Applications*, *5*, 163–175.

Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980.

Kumar, P., Mukherjee, S., Saini, R., Kaushik, P., Roy, P., & Dogra, D. P. (2019). Multimodal gait recognition with inertial sensor data and video using evolutionary algorithm. *IEEE Transactions on Fuzzy Systems*, *27*(5), 956–965.

Lee, C. P., Tan, A. W., & Tan, S. C. (2013). Gait recognition via optimally interpolated deformable contours. *Pattern Recognition Letters*, *34*(6), 663–669.

Lee, C. P., Tan, A. W., & Tan, S. C. (2014). Gait probability image: an information-theoretic model of gait representation. *Journal of Visual Communication and Image Representation*, *25*(6), 1489–1492.

Li, X., Makihara, Y., Xu, C., Muramatsu, D., Yagi, Y., & Ren, M. (2016). Gait energy response function for clothing invariant gait recognition. In *Computer vision, Asian conference* (pp. 257–272). IEEE.

Li, X., Makihara, Y., Xu, C., Yagi, Y., & Ren, M. (2019). Joint intensity transformer network for gait recognition robust against clothing and carrying status. *IEEE Transactions on Information Forensics and Security*, *14*(12), 3102–3115.

Li, X., Makihara, Y., Xu, C., Yagi, Y., & Ren, M. (2020). Gait recognition invariant to carried objects using alpha blending generative adversarial networks. *Pattern Recognition*, *105*, 1–12.

Li, H., Qiu, Y., Zhao, H., Zhan, J., Chen, R., Wei, T., et al. (2022). GaitSlice: A Gait recognition model based on spatio-temporal slice features. *Pattern Recognition*, *124*, Article 108453.

Liao, R., Yu, S., An, W., & Huang, Y. (2020). A model-based gait recognition method with body pose and human prior knowledge. *Pattern Recognition*, *98*, Article 107069.

Liu, D., Ye, M., Li, X., Zhang, F., & Lin, L. (2016). Memory-based gait recognition. In *Machine vision, British conference* (pp. 82.1–82.12). BMVA Press.

Liu, X., You, Z., He, Y., Bi, S., & Wang, J. (2022). Symmetry-Driven hyper feature GCN for skeleton-based gait recognition. *Pattern Recognition*, *125*, Article 108520.

Liu, W., Zhang, C., Ma, H., & Li, S. (2018). Learning efficient spatial–temporal gait features with deep learning for human identification. *Neuroinformatics*, *16*, 457–471.

Lynnerup, N., & Larsen, P. (2014). Gait as evidence. *IET Biometrics*, *3*(2), 47–54.

Makihara, Y., Suzuki, A., Muramatsu, D., Li, X., & Yagi, Y. (2017). Joint intensity and spatial metric learning for robust gait recognition. In *Computer vision and pattern recognition, 30th IEEE conference* (pp. 5705–5715). IEEE.

Mao, J., Xu, W., Yang, Y., Wang, J., Huang, Z., & Yuille, A. (2014). Deep captioning with multimodal recurrent neural networks. arXiv preprint arXiv:1412.6632.

Redmon, J., & Farhadi, A. (2018). YOLOV3: An incremental improvement. arXiv: 1804.02767.

Ren, S., He, K., Girshick, R., & Sun, J. (2015). Faster R-CNN: Towards real-time object detection with region proposal networks. In *Neural information processing systems, Proceedings of the* (pp. 91–99). ACM.

Sadeghzadehyazdi, N., Batabyal, T., & Acton, S. T. (2021). Modeling spatiotemporal patterns of gait anomaly with a CNN-LSTM deep neural network. *Expert Systems with Applications*, *185*, Article 115582.

Schuster, M., & Paliwal, K. K. (1997). Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing*, *45*, 2673–2681.

Shiraga, K., Makihara, Y., Muramatsu, D., Echigo, T., & Yagi, Y. (2016). Geinet: View-invariant gait recognition using a convolutional neural network. In *Biometrics, 8th IAPR international conference* (pp. 1–8). IEEE.

Takemura, N., Makihara, Y., Muramatsu, D., Echigo, T., & Yagi, Y. (2019). On input/output architectures for convolutional neural network-based cross-view gait recognition. *IEEE Transactions on Circuits and Systems for Video Technology*, *29*(9), 2708–2719.

Uddin, M. Z., Ngo, T. T., Makihara, Y., Takemura, N., Li, X., Muramatsu, D., et al. (2018). The OU-ISIR large population gait database with real-life carried object and its performance evaluation. *IPSJ Transactions on Computer Vision and Applications*, *10*(5), 1–11.

Wang, X., & Yan, W. Q. (2020). Human gait recognition based on frame-by-frame gait energy images and convolutional long short-term memory. *International Journal of Neural Systems*, *30*(01), Article 1950027.

Whytock, T., Belyaev, A., & Robertson, N. M. (2015). On covariate factor detection and removal for robust gait recognition. *Machine Vision and Applications*, *26*(5), 661–674.

Wu, Z., Huang, Y., Wang, L., Wang, X., & Tan, T. (2017). A comprehensive study on cross-view gait based human identification with deep CNNs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *39*(2), 209–226.

Xu, D., Yan, S., Tao, D., Zhang, L., Li, X., & Zhang, H. J. (2006). Human gait recognition with matrix representation. *IEEE Transactions on Circuits and Systems for Video Technology*, *16*(7), 896–903.

Yu, S., Chen, H., Reyes, E. B. G., & Poh, N. (2017). Gaitgan: invariant gait feature extraction using generative adversarial networks. In *Computer vision and pattern recognition, ieee conference* (pp. 30–37). IEEE.

Yu, S., Liao, R., An, W., Chen, H., Garcia, E. B., Huang, Y., et al. (2019). Gaitganv2: invariant gait feature extraction using generative adversarial networks. *Pattern Recognition*, *87*, 179–189.

Yu, S., Tan, D., & Tan, T. (2006). A framework for evaluating the effect of view angle, clothing and carrying condition on gait recognition. In *Pattern recognition, 18th international conference* (pp. 441–444). IEEE.

Zeiler, M., & Fergus, R. (2014). Visualizing and understanding convolutional networks. In *Computer vision, European conference* (pp. 818–833). Springer.

Zhang, Y., Huang, Y., Wang, L., & Yu, S. (2019). A comprehensive study on gait biometrics using a joint CNN-based method. *Pattern Recognition*, *93*, 228–236.

Zhang, K., Luo, W., Ma, L., Liu, W., & Li, H. (2019). Learning joint gait representation via quintuplet loss minimization. In *Computer vision and pattern recognition, IEEE conference* (pp. 4700–4709). IEEE.

Zhang, Z., Tran, L. V., Yin, X., Atoum, Y., Liu, X., Wan, J. W., et al. (2019). Gait recognition via disentangled representation learning. In *Computer vision and pattern recognition, IEEE conference* (pp. 4710–4719). IEEE.