Multimedia Tools and Applications
https://doi.org/10.1007/s11042-024-18630-0

Human crowd behaviour analysis based on video segmentation and classification using expectation–maximization with deep learning architectures

Shruti Garg[1] · Sudhir Sharma[2] · Sumit Dhariwal[3] · W. Deva Priya[4] · Mangal Singh[5] · S. Ramesh[6]

Received: 14 September 2023 / Revised: 8 February 2024 / Accepted: 12 February 2024

Abstract
In recent years, the demand for automatic crowd behavior analysis has surged, driven by the need to ensure public safety and minimize casualties during events of public and religious significance. However, effectively analyzing the nonlinearities present in real-world crowd images and videos remains a challenge. To address this, research proposes a novel approach leveraging deep learning (DL) architectures for the segmentation and classification of human crowd behavior. Our method begins by collecting input from surveillance videos capturing crowd activity, which is then processed to remove noise and extract the crowd scene. Subsequently, we employ an expectation–maximization-based ZFNet architecture for accurate video segmentation. The segmented video is then classified using transfer exponential Conjugate Gradient Neural Networks, enhancing the precision of crowd behavior characterization. Our method has been proven effective in experimental analysis on many human crowd datasets, with significant results of average mean precision (MAP) of 59%, the mean square error (MSE) of 61%, accuracy in the training of 95%, validation precision of 95%, and selectivity of 88%. The potential of DL-based methods to advance crowd behavior analysis for improved privacy and security is highlighted by this study.

Keywords Human crowd · Behavior analysis · ZFNet architecture · Conjugate gradient · Expectation–maximization

1 Introduction

The behaviours or actions of a group of people who have assembled for a brief time while paying attention to a specific item or event. A common component of many human endeavors is crowdedness. Every day, many pedestrians are handled in transport hubs, tall buildings, stadiums, and other public places. Effective crowd control is crucial for maintaining safety in these situations and determining one's quality of life. Fires, crowd violence, or the ecstasy of a few crowd members are only a few examples of crowd

Extended author information available on the last page of the article

tragedies, in which people are seriously injured or killed as a result of being crushed or trampled. Such incidents can and have happened during rock concerts, religious services, and athletic events [1]. During the admission, occupation, and evacuation of something like a public event facility, serious injury and disease can occur. Because there are so many cameras available now that make it easy to record and save video, video surveillance of individuals is an often used technology. The majority of these tools rely on a user to review the material that has been stored and interpret its content. Given this restriction, it is vital to offer video surveillance systems that enable automatic behaviour recognition [2]. Computer vision techniques can be used to implement these kinds of systems because they make it possible to recognize unsupervised patterns of human activity, such as gestures, movements, and other activities. Numerous studies are being done right now on human behaviour analysis, like [3], which have helped to identify different forms of human behaviour in video clips. Taking into account their range in time from seconds to hours, these behaviours have been ranked from the most basic to the most sophisticated. When taking into account these Closed Circuit Television (CCTV) cameras and other installation systems, automated crowd research plays a significant part in crowd analysis and visual surveillance recordings [4]. Designing public areas, visual surveillance systems, and intelligently managed physical environments is so important. These kinds of systems will have many useful uses, such as crowd flow monitoring, accident management, and coordinating evacuation plans necessary in the unfortunate case of a sudden and uncontrolled fire or the presence of riots in urban areas in particular [5]. Researchers have looked into the situation of acquiring motion data at a higher level in the research paperwork. This indicates that the motion information does not account for specific moving or stationary objects. As a result, these techniques frequently require a variety of features, such as multi-resolution histograms, spatiotemporal cuboids, appearance or motion descriptors, and spatiotemporal cubes [6].

The contribution of this research is as follows: This research presents a novel approach to human crowd behaviour analysis by integrating segmentation and classification through deep learning architectures. Unlike existing methods, our proposed technique utilizes an expectation–maximization-based ZFNet architecture for video scene segmentation, enabling more accurate delineation of crowd dynamics. Additionally, we introduce transfer exponential conjugate gradient neural networks for classification, enhancing the precision of crowd behaviour characterization. By seamlessly integrating these two components, our method offers a comprehensive and effective solution for understanding complex crowd behaviours in surveillance videos. This novel methodology advances the latest developments in human crowd analysis by improving classification performance as well as segmentation accuracy.

The remaining research is organized as follows: Section 2 contrasts and compares previous studies on the topic. In Section 3, an in-depth description of the ZFNet the building's expectation–maximization-based video segmentation method is given. Transfers exponential Conjugate gradient neural networks are then used for data categorization. The experimental analysis carried out for this study is presented in Section 4. In the fifth section, we wrap up the study's main findings and talk about possible directions for further research.

2 Related works

Crowd safety in public places has always been a serious but difficult issue, especially in high-density gathering areas. The higher the crowd level, the easier it is to lose control [7], which can result in severe casualties. To aid in mitigation and decision-making, it is important to search for an intelligent form of crowd analysis in public areas. Crowd counting and density estimation are valuable components of crowd analysis [8] since they can help measure the importance of activities and provide appropriate staff with information to aid decision-making. As a result, crowd counting and density estimation have become hot topics in the security sector, with applications ranging from video surveillance to traffic control to public safety and urban planning [9]. Numerous crowd-analysis articles were examined in the work [10]. The two main subfields of crowd analysis are statistics and behaviour. Anomaly detection is frequently discussed in crowd behaviour analysis. Any subtopic of crowd behaviour analysis can experience anomalies. Finding unknown or understudied crowd analysis sub-areas that could profit from DL is the goal of this project. The author of [11] studied the crowd-related literature, including techniques for behaviour analysis and crowd surveillance. The author also provided descriptions of the methodology and datasets used. Different techniques and current deep learning concepts have been assessed. The various contemporary methods for crowd monitoring and analysis are explained in this text. The study [12] suggested a picture classification, crowd management, and warning system for the Hajj. Images are classified using Convolutional neural network models (CNN), a DL (deep learning) technology. CNN has found various uses in the scientific and industrial domains, including speech recognition and image categorization. The author [13] suggests the Density density-independent and Scale Aware Model (DISAM), which works well for high-density crowds where photographs only show a portion of the human head. CNN is used to generate a reply matrix utilizing scale-aware head suggestions and it is also used as a head detector to ascertain the odds of a skull in an image. The "you only look once" (YOLO) detection technique is commonly used to locate objects in photos with a significant amount of perspective values, or minimum threshold values, according to [14]. In order to create multipolar adjusted maps of density for crowd counting, work [15] suggested using CNN and learning to scale. It generates a patch-level density map by a density estimation process, which it then classifies into various densities. For each patch densities map, a method for online learning for centers with multi-polar loss is applied. In [16], CNN as well as short-term memory are utilized to calculate crowd density in surveillance videos. For estimating crowd density [19], two traditional Googlenet [17] and VGGNet [18], were utilized. Similar to this, [20] first estimates the size of the crowd in general, and then counts the precise number of persons present. The accuracy of 90% is still maintained by the efficiency. To find and keep an eye on a person in a crowded area, localization information might be employed [21]. We have built a regression-guided detecting network (RDNet) for RGB-Datasets that concurrently estimates head counts and uses boundaries to localize heads in images. Similar to [22], an accurate localization of the heads in a dense image was achieved using a density map. Using the neural network, localization was discovered in [23] with the aid of a statistic called Mean Localization Error (MLE) [24]. Employed image processing to determine crowd behavior using optical flow as well as motion history image techniques. As in [25], the identification of abnormal behavior was achieved by the use of a Support Vector Machine (SVM) in conjunction with an optical circulation technique. In [26], a Cascades Shallow Auto Encoder (CDA) and a combination of multi-frame optical flow information are presented to identify crowd behavior. Isometrically projection (ISOMAP), spatiotemporal, and temporal texture models were used to identify abnormal crowds. Table 1 explains the overview of related works.

3 Proposed system

In this section, the proposed model for video segmentation and classification in human crowd analysis harnesses the power of deep learning (DL) techniques to comprehensively analyze crowd behavior from surveillance footage. The process initiates with input collection, where surveillance video undergoes noise removal to ensure clarity, followed by obtaining the crowd scene for analysis [27]. Segmentation, the pivotal stage, employs an expectation–maximization-based ZFNet architecture to precisely delineate individual elements within the crowd scene, facilitating the identification of specific behaviors and interactions.

Subsequently, the segmented video segments are fed into a transfer exponential Conjugate gradient neural network (NN) for classification. This particular neural network improves performance and resilience by generalizing information gained from models that have been trained across a variety of crowd environments and settings by utilizing methods of transfer learning [28]. Three main elements make up the suggested structure, which is shown in Fig. 1: input processing, segmentation, and classification. The input processing stage employs an individual activity recognition chain to extract features from sensor signals, converting them into time series data representing behavioural primitives or quantitative user behaviour characteristics [29]. Segmentation involves dividing each frame scene into non-overlapping cubes and extracting global and local descriptors. Local descriptors, crucial for capturing fine-grained details within the crowd scene, utilize the Inner Temporal Approach (ITA) and a space–time neighborhood approach to assess the similarity between patches. The local descriptor, a kind of local patch descriptor, determines how similar patches are by using the Structural Similarity Index Method (SSIM) approach. Regarding the first local description, each patch's space–time neighbourhood sections consist of one for the spatial neighbourhood, which includes the patch itself in the center, and one for the temporal neighbourhood, which comes after the patch. The initial local descriptor [d0,…, d9] gives rise to the SSIM values. In terms of the TIA, the SSIM value is calculated as [D0…, Dt-1] for each frame in the patch. Finally, the combined SSIM values from the two approaches are used to create the local descriptor [d0,…, d9, D0,…, Dt-1].

3.1 Expectation–maximization-based ZFNet architecture in video segmentation

The Expectation–Maximization technique was used to fit the WMM, as is customary, I considered it incomplete and is supplemented with a g dimensional z b, where z = 1 is true if r i I come from the kth component and 0 otherwise. Component memberships are defined as realizations of random vectors $z_1, z_2, \ldots, z_n)$ dispersed unconditionally according to the Mulr multinomial distribution $\left(1, \pi_1, \ldots, \pi_k\right)$. The EM iteratively maximizes the conditional expectation $Q\left(\Psi : \hat{\Psi}^n\right)$ of the complete-data log-likelihood for observed data v in (1,2) concerning the observed data v given an estimate $\overbrace{\text{Wb}}^{\Psi^{n+4}}$ for the parameters.

$$\begin{aligned} &= \sum_{i=1}^{s} E_{\hat{\gamma} i}\left[\log\left\{f\left(z_i \Psi\right) f\left(\Gamma_i \middle| Z_i \Psi^i\right)\right\} \Gamma_i\right] \\ &= \sum_{i}^{s} \sum_{k=1}^{g} E_{\hat{p}n} \left|Z_k\right| \Gamma_i \log\left(\left\{\dot{\pi}_k^{(i)} \rho W\left(\Gamma_i \widehat{\sum}_k^{(i)}, \hat{n}_k^{(j)}\right)\right\}\right) \end{aligned} \tag{1}$$

$$= \frac{\dot{\pi}_i^n f w\left(r_i \hat{2}_k^{(i)}, \hat{n}_i^{(n)}\right)}{\sum_{i-1}^{*} \hat{n}_i^{(n)} f w\left(r_i \hat{\Sigma}_i^n, \hat{n}_i^{(t)}\right)} \tag{2}$$

Table 1 Summary of related work

Ref	Methodology	Purpose	Results
[10]	Statistics and behaviour analysis	Identify anomalies in crowd behaviour	Examined various sub-areas of crowd behaviour analysis to identify potential applications of deep learning (DL)
[12]	Convolutional Neural Network (CNN) for image classification	Propose image categorization, crowd control, and warning system for Hajj	Implemented CNN for image classification and proposed a system for crowd management
[13]	Density Independent and Scale Aware Model (DISAM)	Develop a model for crowd density estimation	DISAM model performs well for high-density crowds with partially visible heads in images
[14]	"You Only Look Once" (YOLO) detection method	Object detection in images with perspective distortion	Effectively detects objects in images with varying perspectives
[15]	CNN and learn to scale for multipolar normalized density maps	Generate density maps for crowd-counting	Used CNN and scaling techniques to produce multipolar normalized density maps for crowd-counting
[16]	CNN and short-term memory for crowd density calculation	Estimate crowd density in surveillance videos	CNN and short-term memory networks used to estimate crowd density in videos
[17]	Traditional Googlenet	Estimate crowd density	Googlenet utilized for crowd density estimation
[18]	VGGNet	Estimate crowd density	VGGNet employed for crowd density estimation
[19]	Crowd size estimation and precise counting	Estimate crowd size and count accurately	Achieved 90% accuracy in crowd size estimation and counting
[20]	Utilize localization information for crowd-monitoring	Develop a method for person localization in crowds	Proposed a regression-guided detection network (RDNet) for head localization in images
[21]	Density map for accurate head localization	Achieve accurate head localization in dense images	Employed density map for precise head localization in crowded scenes
[22]	Image processing for crowd behaviour analysis	Determine crowd behaviour using optical flow	Identified anomalous behaviour using optical flow and SVM
[23]	Neural network for localization	Utilize Mean Localization Error (MLE) for localization	Used MLE statistic for accurate person localization in crowded scenes
[24]	Image processing for crowd behaviour analysis	Analyze crowd behaviour using optical flow and motion history images	Employed optical flow and motion history images for crowd behaviour analysis
[25]	Anomaly detection using SVM	Identify anomalous behaviour in crowds	Detected anomalous behaviour using SVM and optical flow techniques
[26]	Cascade Deep Auto Encoder (CDA) for crowd activity detection	Detect crowd activity using multi-frame optical flow	Proposed CDA and multi-frame optical flow for anomalous crowd activity detection

As it represents an estimate of the posterior probability z_i^n that Γ_i belongs to a kth component of mixture under a given parameter set $\hat{\psi}$.The algorithm's maximum stage aims to increase $Q\left(\Psi^*\hat{\Psi}^n\right)$ by Eq. (3) to obtain a fresh parameter estimate $\hat{\Psi}(t+1)$.

$$\hat{\Psi}^{(i+1)} = \text{argmax } Q\left(\Psi^*\hat{\Psi}^{(n}\right) \tag{3}$$

By maximising $Q\left(\Psi_{;}\hat{\Psi}^*\right)$ with the restriction $\sum_{i=1}^{\pi}\pi_k^{\pi+1)} = 1$, the new estimates $\dot{\pi}_k^{n+1)}$ for π_k are produced via update rule via Eq. (4)

$$\pi_k^{(N+1)} = \frac{1}{N}\sum_{i=1}^{N} z_k^n \tag{4}$$

By utilizing a few matrix derivation techniques. By using Eq. (5), we can get the updated equations for various parameters.

$$\begin{aligned} &= \sum_{i=1}^{N} \frac{\partial}{\partial\Sigma_i}\left(z_u^* \log\left\{\tilde{t}_k^n f_w\left(r_i, z_i, m_k\right)\right\}\right) \\ &= \sum_{i=1}^{N} Z_i^n \frac{\partial}{\partial\Sigma_k}\log f_w\left(r_i : \Sigma_i, n_i\right) \\ &= \sum_{k=1}^{N} z_k^N\left(\frac{1}{2}E_k^{-1}r_i\Sigma_k^{-1} - \frac{n_k}{2}E_i^{-1}\right) \end{aligned} \tag{5}$$

After premultiplying the previous equation by 2, we obtain the following for all k by Eq. (6):

$$\frac{\partial}{\partial\Sigma_k}Q\left(\Psi_{;}\overline{\mathbf{T}}^n\right) = 0 \approx \Sigma_k^{p+1)} = \frac{\sum_{i=1}^{n}\overline{z}_a^n\Gamma_1}{\sum_{*-1}^{n}\hat{z}_k^n n_k} \tag{6}$$

Equation (7) is solved numerically to estimate $n_{N'}$, which is then reintroduced into (7) to obtain a suitable value for $Q\left(\nabla_{:}\hat{\psi}'^i\right)$. This is comparable to solving the following Eq. (7) separately for each component

$$\sum_{i=1}^{N} z_n^n \log\left|\frac{r_i z_k}{2}\right| = \sum_{i=1}^{N} z_i^n \sum_{j=1}^{n}\psi\left(\frac{1}{2}\left(n_k - j + 1\right)\right) \tag{7}$$

where the digamma function, ψi^i is represented by the letter Σi_{is} in Eq. (6). Then, formula (7) is solved numerically in a small number of iterations, and the solution n_e^{0+1} is reintroduced in (7) to have a suitable value for 2_i^{n+11}. In training the segmentation network, we address class imbalance by employing various loss functions, including cross-entropy, commonly used for segmenting medical images. Equation (7) calculates the cross-entropy loss, averaging pixel predictions, but it may lead to errors with unbalanced class representation. To mitigate bias towards wider classes, we resample the data space. Optimization techniques involve minimizing the chosen loss function using backpropagation. The ZFNet architecture, depicted in Fig. 2, guides the network's training process. Regularization techniques such as dropout and batch normalization are applied to prevent overfitting. Additionally, we utilize techniques like stochastic gradient descent (SGD) or Adam optimizer for efficient convergenceTechniques like grid search and random searches are used to tweak the algorithm's hyperparameters which include its rate of learning and batch size. Early stopping is employed to prevent overfitting, while model performance is monitored using validation data. Finally, the trained model's performance is evaluated on unseen test data to ensure generalization capability [30].

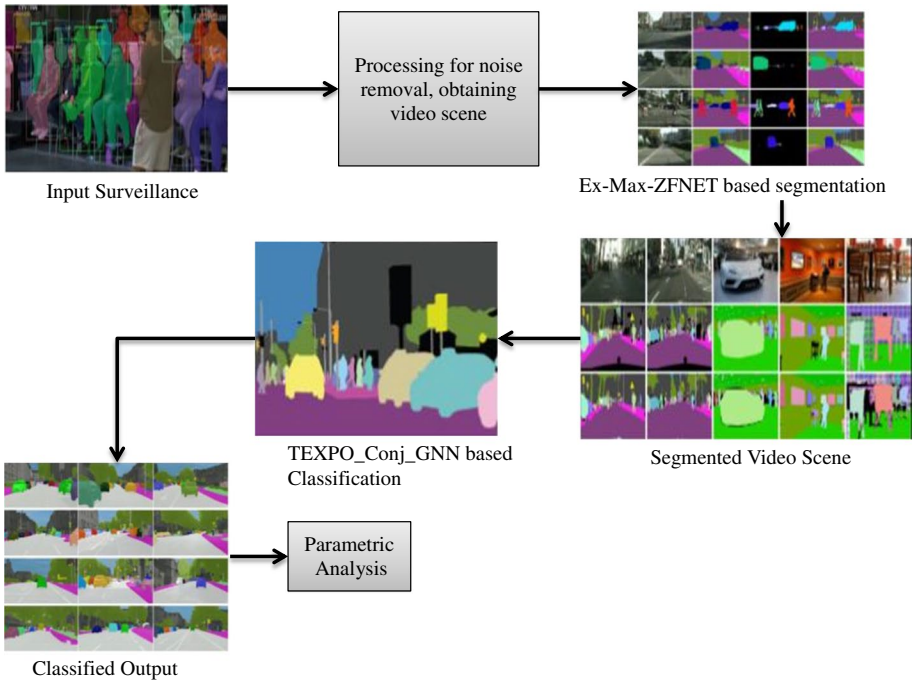

Fig. 1 Proposed architecture

A lower number suggests a tighter connection between the same object sections in multiple photos and better consistency in the change caused by the masking procedure. Utilizing features from layers l = 5 and l = 7, we compare scoresΔ for the left eye, right eye, and nose to random areas of the object. The layer 5 features' lower scores for these regions compared to random object regions demonstrate that the model does build some degree of correlation [31, 32].

3.2 Transfer exponential Conjugate gradient neural networks-based classification

The input data for our neural network model consists of images with three layers: height (h), width (w), and depth (d), where d represents the feature or channel dimension, and h and w represent the spatial dimensions. The input layer has dimensions h × w and d color channels (d = 1 for grayscale or d = 3 for RGB). Equation (8) describes how the vector output y_{ij} is calculated from the input vector x_{ij} at position, i use a function f_{ks}.

$$y_{ij} = f_{ks}\left(\left\{x_{ij} + \lhd isj + sj\right\}, 0 \le \delta_{i,}\delta_j \le k\right) \tag{8}$$

To reduce the parameter count, we employ Eq. (9) to define $(I_n(g))$ the average of a function g over a collection of independent random variables $g(\boldsymbol{x}_i)$.

$$I_n(g) = \frac{1}{n}\sum_{i=1}^{n} g(\boldsymbol{x}_i) \tag{9}$$

Then a simple evaluation gives us $\mathbb{E}(I(g) - I_n(g))^2 = \frac{\mathrm{Var}(g)}{n}$, $\mathrm{Var}(g) = \int_X g^2(\boldsymbol{x})dx - \left(\int_X g(\boldsymbol{x})d\boldsymbol{x}\right)^2$

Given that the neural network (NN) is composed of three layers: input, result, and hidden ($e^{(n)}$), It is required to calculate the result of the layer that is concealed prior to calculating the output of the whole network. Equation (9), where indicates activation function, $\vec{i}$ represents the hidden neuron, denotes the input neurons, and $\underset{(ai)}{mm}$ is utilized to determine the hidden layer's output, or ein, and denotes bias weight. The NN model is given (10).

$$\begin{aligned} e^{(n)} &= nf\left(w_{(ni)}^{(m)} + \sum_{j=1}^{n} w_{(j)}^{(N)} F_D\right) \\ \hat{\sigma}_{\hat{o}} &= nf\left(w_{(\dot{\omega}o)} + \sum_{i=1}^{s} w_{\left(\hat{i}\right)}^{(\omega)} e^{(m)}\right) \end{aligned} \tag{10}$$

The weight matrices are provided in (9) and (10). Equation (11) is utilized to generate weight matrices and biases for optimization, where W_n represents the weight matrix and B_n the bias value.

$$\begin{aligned} W_n = U_n &= \sum_{m=1}^{N} a \cdot \left(\text{rand} - \frac{1}{2}\right). \\ B_n &= \sum_{n=1}^{N} a \cdot \left(\text{rand} - \frac{1}{2}\right). \\ \left|\mathcal{R}(f) - \widehat{\mathcal{R}}_n(f)\right| &\leq \sup_{f \in \mathcal{H}_m} \left|\mathcal{R}(f) - \widehat{\mathcal{R}}_n(f)\right| = \sup_{f \in \mathcal{H}_m} \left|I(g) - I_n(g)\right| \end{aligned} \tag{11}$$

where $W_n = N$ weight within the weight matrix. The term "rand" refers to the number chosen at random in (1) that is between [0,1], where B_n is a bias value and an is a constant parameter for the suggested technique that is less than 1. As such, formula (12) gives the weight list matrix:

$$W^c = \left[W_n^1, W_n^2, W_n^3, \ldots, W_n^{N-1}\right] \tag{12}$$

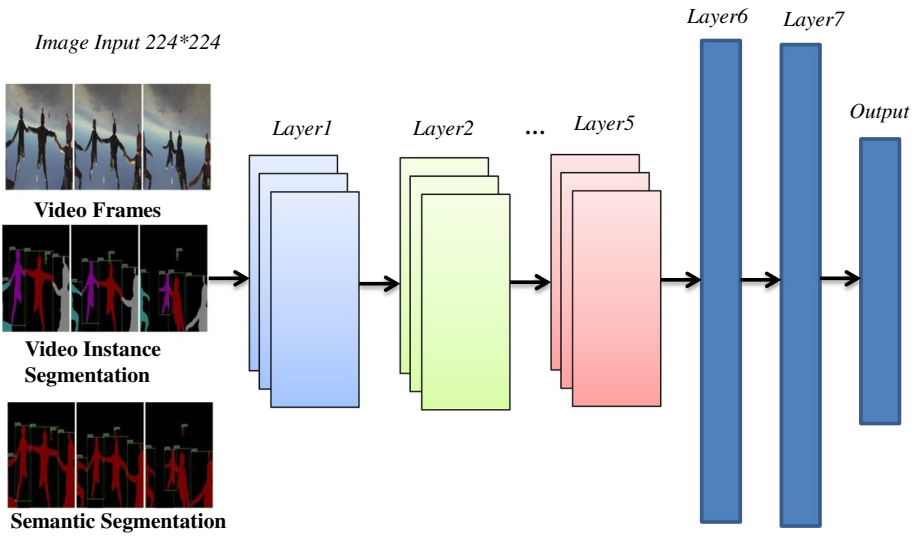

Fig. 2 Architecture of ZFNet

Weight matrices are organized into a weight list matrix as shown in Eq. (12). The neural network process predicts the total of square errors for every weight matrix. A layer for input, a hidden or "state" layer, and a result layer comprise the three-layer network framework. Equations (13) and (14) describe the propagation of input vectors through weight layers in both simple recurrent networks and neural networks.

$$\begin{aligned} \mathrm{net}_j(t) &= \sum_i^n x_i(t) w_{m(j)} + B_{m(j)} \\ \inf_{m \in K_m} || f^* - f_m ||^2_{L^2(P)} &\lesssim \frac{\Delta(f^*)^2}{m} \end{aligned} \tag{13}$$

where, $B_{m(j)}$ is a bias and m is a number of inputs. In a basic recurrent network, an input vector is similarly transmitted across a weight layer. but it is also paired with the activation of the previous state by a second recurrent weight layer, U by Eq. (14).

$$\begin{aligned} & y_j(t) = f\left(\mathrm{net}_j(t)\right). \\ & \mathrm{net}_j(t) = \sum_i^{\sum_i} x_i(t) W_{s(m)} + \sum_i^n n(t-1) U_{n(j)} + B_{m(j)}, \\ & y_j(t) = f\left(\mathrm{net}_j(t)\right), \\ & \Delta(f) := \inf_j \int_{\mathbb{R}^d} \| \omega \|_1 \left| \widehat{f}(\omega) \right| d\omega < \infty, \end{aligned} \tag{14}$$

where f is an extension of f to $\inf_j \int_{\mathbb{R}^d}$ Fourier transform. In (14) the convergence rate is dimension-independent. However, because it uses the Fourier transform, constant $\Delta(f*)$ could be dimension-dependent. In both cases, the state and a set of weights for output W generated by eq control the output of the network (15).

$$\begin{aligned} et_k\,(t) &= \textstyle\sum_j^M y_j\,(t)\, W_{mak\,j} + B_{m|k,} \\ Y_k\,(t) &= g\,\left(net_k\,(t)\right), \end{aligned} \tag{15}$$

g is an output function. Thus, the error is determined using Eq. (16):

$$E = \left(T_k - Y_k\right) \tag{16}$$

Equation (17) gives the network's performance index:

$$\begin{aligned} V(x) &= \frac{1}{2} \sum_{k=1}^K \left(T_k - X_k\right)^T \left(T_k - Y_k\right) \\ V_F(x) &= \frac{1}{2} \sum_{k=1}^K E^T \cdot E. \end{aligned} \tag{17}$$

$$V_\mu(x) = \frac{\sum_{j=1}^N V_F(x)}{P_i} \tag{18}$$

Equation (19) introduces a random feature method,

$$f_m(\boldsymbol{x};\boldsymbol{a}) = \frac{1}{m}\sum_{j=1}^{m} a_j \phi\left(\boldsymbol{x};\boldsymbol{w}_j^0\right) \tag{19}$$

where the i.i.d random variables $\boldsymbol{w}_j^0$ and $\{\boldsymbol{w}_j^0\}$mj = 1 are selected from the prefixed distribution 0 The coefficients are $a = (a1, \ldots, am)T \in R$m and the collection is $\{\varphi(;;$w0j$)\}$ are the random characteristics. The replicating kernel Hilbert space (RKHS), which is caused by the kernel by eq, is the natural function space for this paradigm (20)

$$k\left(\boldsymbol{x},\boldsymbol{x}'\right) = \mathbb{E}_{\boldsymbol{w}\sim\pi_0}\left[\phi(\boldsymbol{x};\boldsymbol{w})\phi\left(\boldsymbol{x}';\boldsymbol{w}\right)\right] \tag{20}$$

Denote by Hk this RKHS. Then for any $f \in$ Hk, there exists $a(\cdot) \in$ L2(π0) such thateq (21).

$$\begin{aligned} f(\boldsymbol{x}) &= \int a(\boldsymbol{w})\phi(\boldsymbol{x};\boldsymbol{w})d\pi_0(\boldsymbol{w}), \\ \|f\|_{\mathcal{H}_k}^2 &= \inf_{a\in S_f}\int a^2(\boldsymbol{w})d\pi_0(\boldsymbol{w}), \end{aligned} \tag{21}$$

In batch-wise training, variations originate from the gradient variance. The noisy gradient is a drawback of using a random sample, but it has the benefit of requiring far fewer calculations per iteration. Please be aware that the rate of convergence in the preceding paragraph is calculated through iterations. To look at the training dynamics of every iteration, we need to first establish the Lyapunov function using Eq. (22).

$$h_t = \|\mathbf{w}^t - \mathbf{w}^*\|_2^2 \tag{22}$$

The formula calculates the separation between the existing solution, $\mathbf{w}^t$, and the ideal solution, $\mathbf{w}^*$ where h_t is a random variable. As a result, using Eq. (23), one can determine the SGD's convergence rate:

$$\begin{aligned} h_{t+1} - h_t &= \|\mathbf{w}^{t+1} - \mathbf{w}^*\|_2^2 - \|\mathbf{w}^t - \mathbf{w}^*\|_2^2 \\ &= \left(\mathbf{w}^{t+1} + \mathbf{w}^{\mathbf{t}} - 2\mathbf{w}^*\right)\left(\mathbf{w}^{t+1} - \mathbf{w}^{\mathbf{t}}\right) \\ &= \left(2\mathbf{w}^{\mathbf{t}} - 2\mathbf{w}^* - \eta_t\nabla\psi_{\mathbf{w}}\left(\mathbf{d}_{\mathbf{t}}\right)\right)\left(-\eta_t\nabla\psi_{\mathbf{w}}\left(\mathbf{d}_{\mathbf{t}}\right)\right) \\ &= -2\eta_t\left(\mathbf{w}^t - \mathbf{w}^*\right)\nabla\psi_{\mathbf{w}}\left(\mathbf{d}_{\mathbf{t}}\right) + \eta_t^2\left(\nabla\psi_{\mathbf{w}}\left(\mathbf{d}_{\mathbf{t}}\right)\right)^2 \end{aligned} \tag{23}$$

It is a random sample of d in the sample space Ω, and the random variable $h_{t+1} - h_t$ depends on the sample drawn ($\mathbf{d}_{\mathbf{t}}$) and the rate of learning (η_t). It indicates the extent to which reducing YAR{∇yw(dt)} improves the convergence rate. We gauge SGD's effectiveness using $(k) = \mathbb{E}\left[\|z(k) - z^*\|^2\right]$, This stands for the expected squared difference between the optimal solution and the solution at time k. Unlike the study for SGD, we will concentrate on two error terms. The first term, called the expected optimization error, defines the expected squared length among z(k) and z*. The average squared distance between the ideal z* and each iterate's z_i (k) is given by Eq. (24).

$$\frac{1}{n}\sum_{i=1}^{n}\mathbb{E}\left[\|z_i(k) - z^*\|^2\right] = \mathbb{E}\left[\|\bar{z}(k) - z^*\|^2\right] + \frac{1}{n}\sum_{i=1}^{n}\mathbb{E}\left[\|z_i(k) - \bar{z}(k)\|^2\right] \tag{24}$$

Thus, comparing the two terms will help us understand how well DSGD is working. indicate by Eq. (25) to simplify the notation.

$$U(k) = \mathbb{E}\left[\|\bar{z}(k) - z^*\|^2\right], V(k) = \sum_{i=1}^{n} \mathbb{E}\left[\|z_i(k) - \bar{z}(k)\|^2\right], \forall k \tag{25}$$

We decided to use Eq. (26) after being motivated by the SGD analysis

$$U(k+1) \leq \left(1 - \frac{1}{k}\right)^2 U(k) + \frac{2L}{\sqrt{n}\mu} \frac{\sqrt{U(k)V(k)}}{k} + \frac{L^2}{n\mu^2} \frac{V(k)}{k^2} + \frac{\sigma^2}{n\mu^2} \frac{1}{k^2} \tag{26}$$

V(k), the observed consensus error, is a reflection of the extra disruptions caused by the differences in solutions. Additionally, Relation (17) demonstrates that U(k)'s predicted convergence rate for SGD cannot be higher than R(k). On the other hand, two more factors will probably become negligible over time if V(k) decays fast enough in relation to U(k), and we would anticipate that U(k) will converge at a rate comparable to R(k) for SGD.

4 Performance analysis

The teaching platform was a Windows 7 64-bit computer equipped with an Intel Xeon E5-1650 processor. The computer resources included CUDA 8.1, Python 2.7, CUDNN 7.5, and Visual Studio by Microsoft 12.0.

Dataset description: The first dataset used for population counts was from UCSD. The information was gathered using a camera that was mounted on a walkway for pedestrians. The dataset comprises 2000 video sequence frames at a resolution of 238×158 pixels, together with ground truth tagging of 49,885 pedestrians per fifth frame. Security cameras installed at a shopping center were used to collect the mall dataset. 2000 frames in all, 320×240 pixels in size. The challenging UCF CC 50 dataset offers a variety of sceneries and densities. This information was collected from a variety of locations, including stadiums, marathons, political rallies, and concerts. There are a total of 50 annotated photos, with 1279 individuals on average per picture. The resolution of each person in this set of images varies from around 94 to 4543, suggesting a broad variation in the image. The limitation on the number of photos available for training and evaluation is a downside of this type of dataset. This dataset's 220 maximum crowd count is too low to accurately assess the counting of highly dense crowds. The 1198 pictures and 330,165 identified heads in the Shanghai Tech collection are available for large-scale crowd counting. In terms of the number of documented heads, this group is among the biggest. The dataset is divided into two categories: Part A and Part B. There are 482 randomly chosen photos from the internet in Part A. Seventeen hundred and sixteen images from an alleyway in Shanghai are included in Part B. UCF-QNRF, which contains 1535 pictures, is the most recent dataset. The range of individuals in this dataset, from 49 to 12,865, results in a significant fluctuation in population density. Moreover, it features crowd videos with a wide range of view sizes and swarm densities, and its enormous resolution of images spans from 400×300 to 9000×6000. The CUHK dataset was gathered in a variety of places, including streets, malls, airports, and parks. 474 video clips from 215 scenes make up the dataset shown in Table 2.

Table 3 shows the analysis of various video datasets based on human crowd behaviour. the datasets compared are UCSD, MALL, UCF_CC_50, World Expo 10, Shanghai Tech

Table 2 Description of datasets

Datasets	Description	No.of images	Resolutions	Min	Ave	Max	Overall count	Accessibility
UCSD	People counting	2000	238×158	11	25	46	49,885	Yes
MALL	People counting	2000	320×240	13	-	53	62,325	Yes
UCF_CC_50	Density estimation	50	Variable	94	1279	4543	63,325	Yes
World Expo 10	Cross-scene crowd counting	3980	576×720	1	50	253	199,923	Yes
Shanghai Tech A, B	Crowd counting	482	Variable	33	501	3139	241,677	Yes
UCF-QNRF	Crowd counting and localization	716	400×300 to 9000×6000	9	123	578	88,488	yes
CUHK	Crowd behaviour	1535	Variable	49	815	12,865	-	Yes

Table 3 Analysis for various video datasets based on human crowd behaviour

Datasets	Techniques	MAP	MSE	Training accuracy	Validation accuracy	Specificity
UCSD	**CNN** [16]	41	38	68	72	65
	SVM [25]	43	42	72	74	68
	HCB_VSC_DLA	44	43	75	77	71
MALL	**CNN**	42	39	72	75	69
	SVM	46	45	73	77	73
	HCB_VSC_DLA	49	47	75	79	75
UCF_CC_50	**CNN**	44	41	74	79	71
	SVM	48	43	78	85	76
	HCB_VSC_DLA	51	48	81	88	77
World Expo 10	**CNN**	45	44	78	81	73
	SVM	49	48	79	83	77
	HCB_VSC_DLA	53	49	83	86	79
Shanghai Tech A, B	**CNN**	47	48	81	83	75
	SVM	49	52	83	88	81
	HCB_VSC_DLA	53	53	85	89	83
UCF-QNRF	**CNN**	49	51	82	85	81
	SVM	51	53	85	89	83
	HCB_VSC_DLA	53	55	88	92	85
CUHK	**CNN**	52	55	84	91	82
	SVM	55	58	89	93	86
	HCB_VSC_DLA	59	61	95	95	88

A, B, UCF-QNRF, CUHK. The parameters analyzed are MAP, MSE, training accuracy, validation accuracy, and specificity.

Figures 3a-e, 4, 5, 6, 7, 8, and 9a-e shows the analysis for various human crowd behaviour datasets. The suggested method shows significant gains in performance measures when compared to current approaches on different datasets of human crowd behavior. The suggested method yielded the following results in the UCSD dataset: the mean squared error (MSE) of 43%, training accuracy of 75%, accuracy for validation of 77%, and sensitivity of 71%. The average mean precision (MAP) was 44%. SVM achieved a MAP of 43%, MSE of 42%, training accuracy of 72%, validation accuracy of 74%, and specificity of 68%, whereas the current CNN earned a MAP of 41%, MSE of 38%, retraining accuracy of 68%, verification accuracy of 72%, and specific of 65%. For comparison. The suggested strategy also performed better in the MALL dataset than the previous approaches, with MAP of 49%, MSE of 47%, trained reliability of 75%, validation precision of 79%, and sensitivity of 75%. MAP of 42%, MSE of 39%, training success of 72%, validation accuracy of 75%, and selectivity of 69% were attained by the current CNN, whereas SVM yielded Gis of 46%, MSE of 45%, learning correctness of 73%, testing accuracy of 77%, and specificity of 73%. Furthermore, the suggested approach scored better on the UCF_CC_50 dataset, showing MAP of 51%, MSE of 48%, training precision of 81%, accuracy for validation of 88%, and specific of 77%. While SVM obtained a MAP of 48%, Msw of 43%, training accuracy of 78%, validation success rate of 85%, and specificity of 76%, the current CNN got a MAP of 44%, MSE of 41%, training quality of 74%, validating accuracy of 79%, and specificity

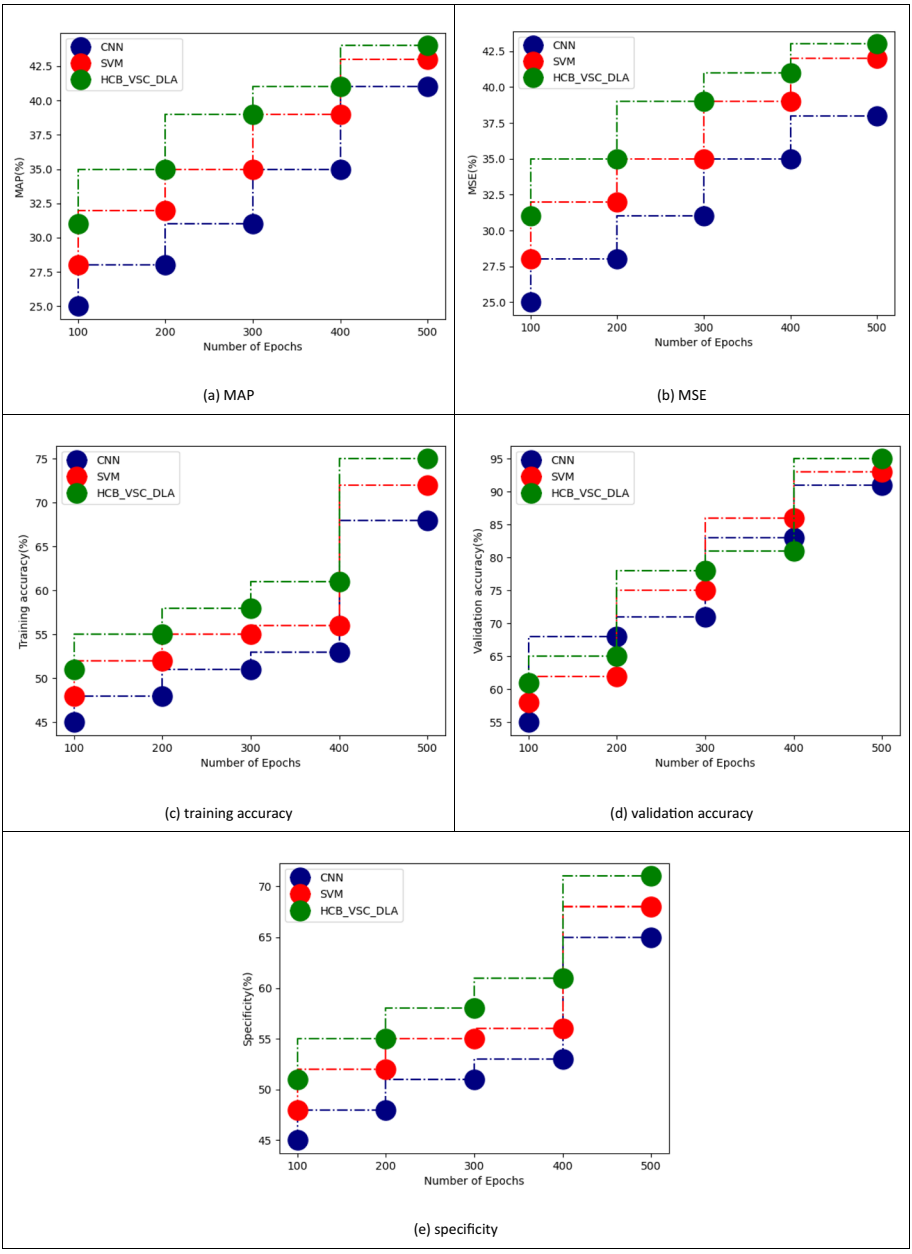

Fig. 3 Examination of the UCSD dataset with respect to (**a**) specificity, (**b**) training accuracy, (**c**) validation accuracy, and (**e**) MAP

of 71%. Furthermore, the approach suggested showed a noteworthy enhancement in the World Expo 10 dataset, exhibiting a MAP of 53%, an MSE of 49%, training precision of 83%, validation precision of 86%, and sensitivity of 79%. MAP of 45%, MSE of 44%, training accuracy of 78%, validation accuracy of 81%, and specificity of 73% were obtained by

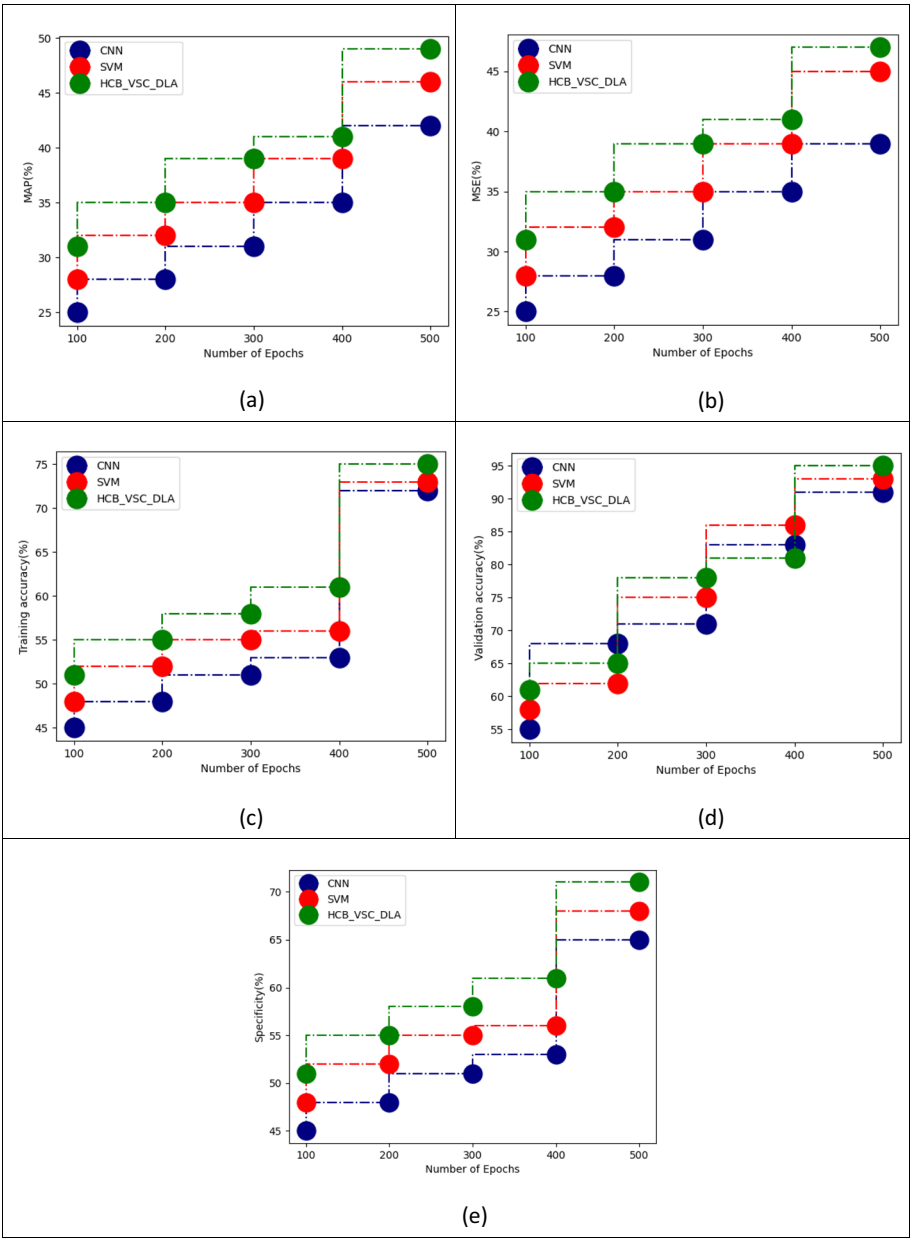

Fig. 4 Analysis of the MALL dataset in terms of (**a**) MAP, (**b**) MSE, (**c**) training accuracy, (**d**) validation accuracy, (**e**) specificity

the current CNN, whereas SVM obtained MAP of 49%, MSE of 48%, trainee accuracy of 79%, testing accuracy of 83%, and specificity of 77%. Additionally, the suggested method demonstrated outstanding outcomes with a MAP of 53%, MSE of 53%, training precision of 85%, validation precision of 89%, and sensitivity of 83% on the Shanghai Tech A, B

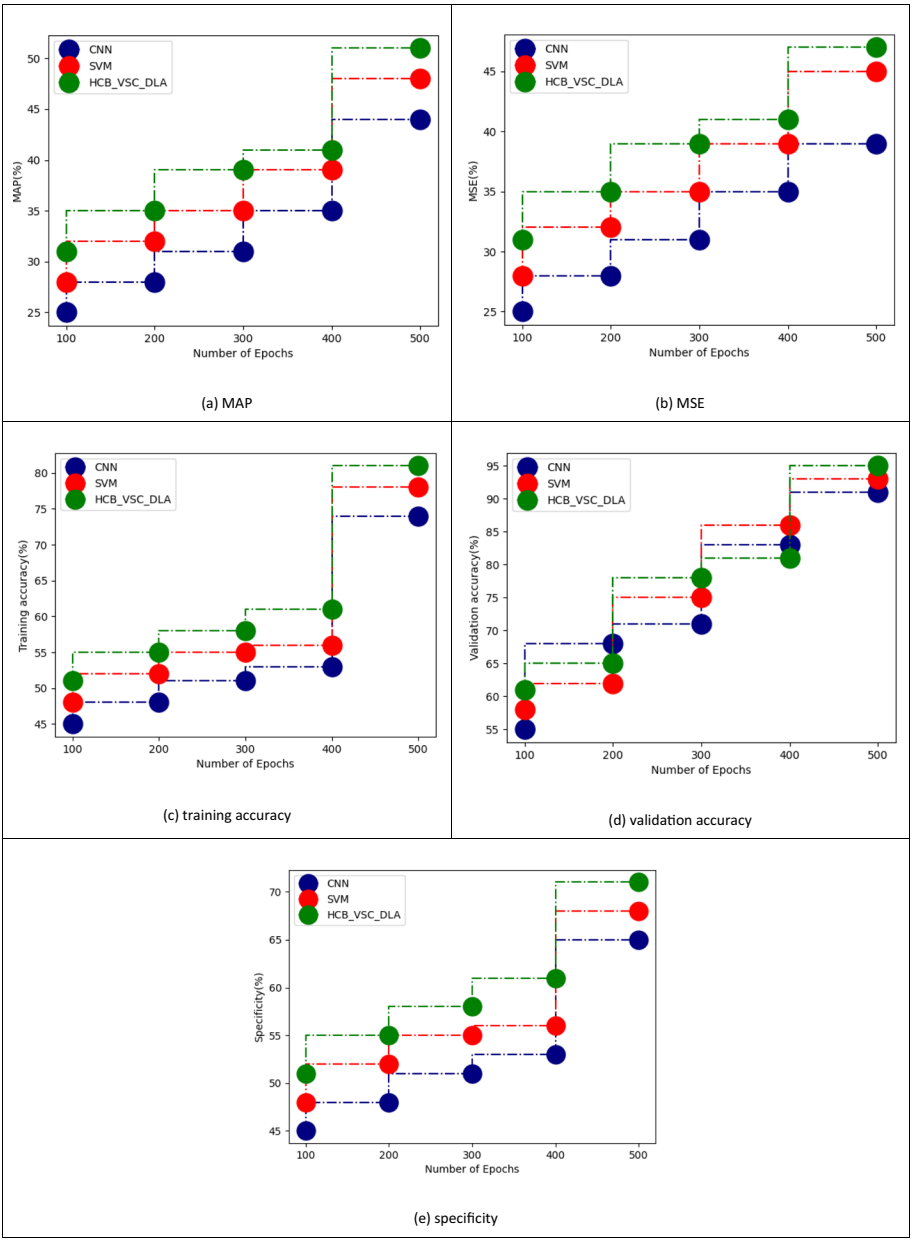

Fig. 5 Examination of the UCF_CC_50dataset concerning (**a**) specificity, (**b**) training precision, (**c**) validation the precision, and (**e**) MSE

dataset. In comparison to the SVM, which obtained a MAP of 49%, MSE of 52%, accuracy in the training of 83%, accuracy for validation of 88%, and specificity of 81%, the current CNN produced MAPs of 47%, 48%, 81%, and specificity of 75%. Finally, the suggested technique demonstrated outstanding results with a MAP of 53%, MSE of 55%, training

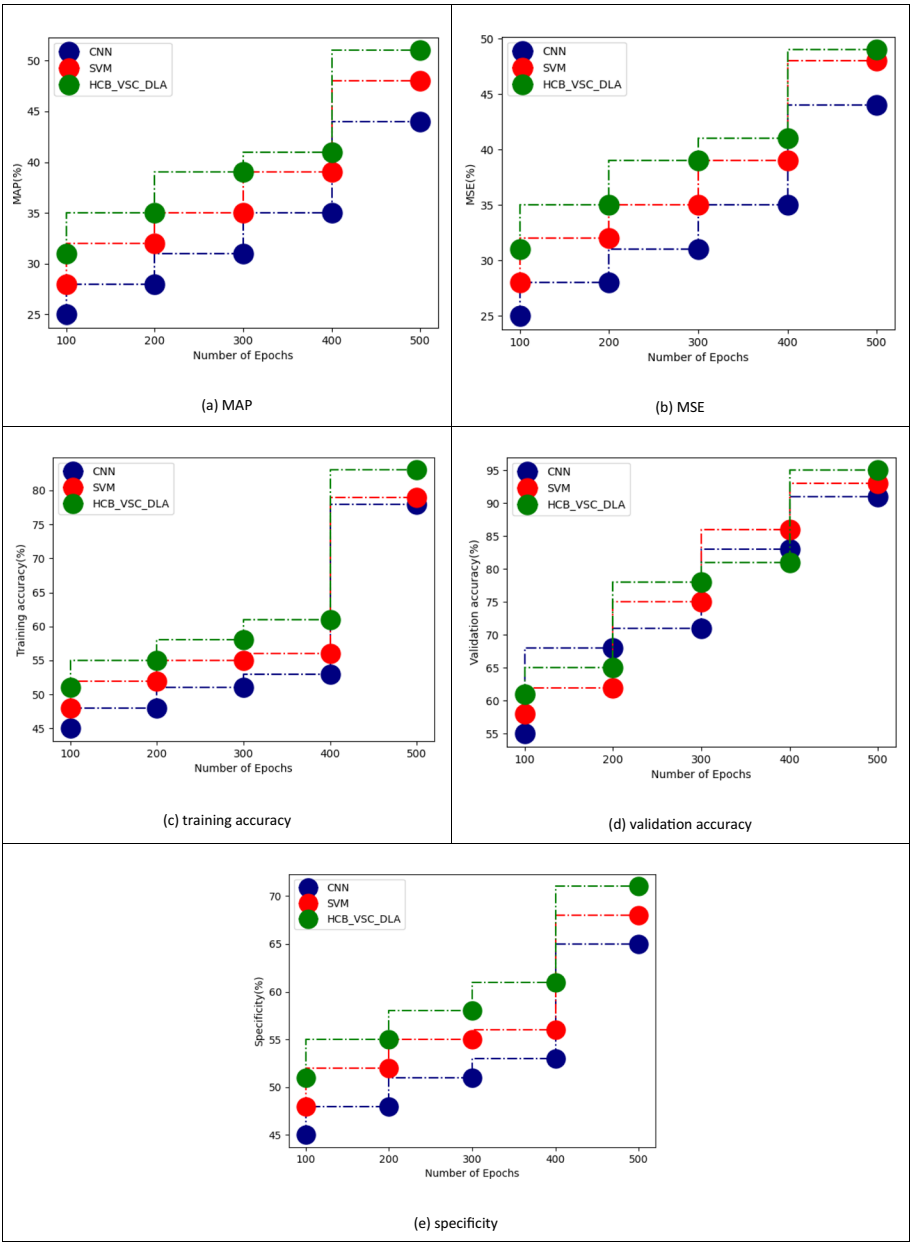

Fig. 6 Examination of the World Expo 10 dataset with respect to (**a**) specificity, (**b**) training accuracy, (**c**) validation accuracy, and (**e**) MSE

reliability of 88%, validation accuracy of 92%, and selectivity of 85% in the UCF-QNRF dataset. MAP of 49%, MSE of 51%, training success rate of 82%, validation success rate of 85%, and selectivity of 81% were acquired by the current CNN, while MAP of 51%, MSE of 53%, train accuracy of 88%, testing accuracy of 92%, and specificity of 85% were

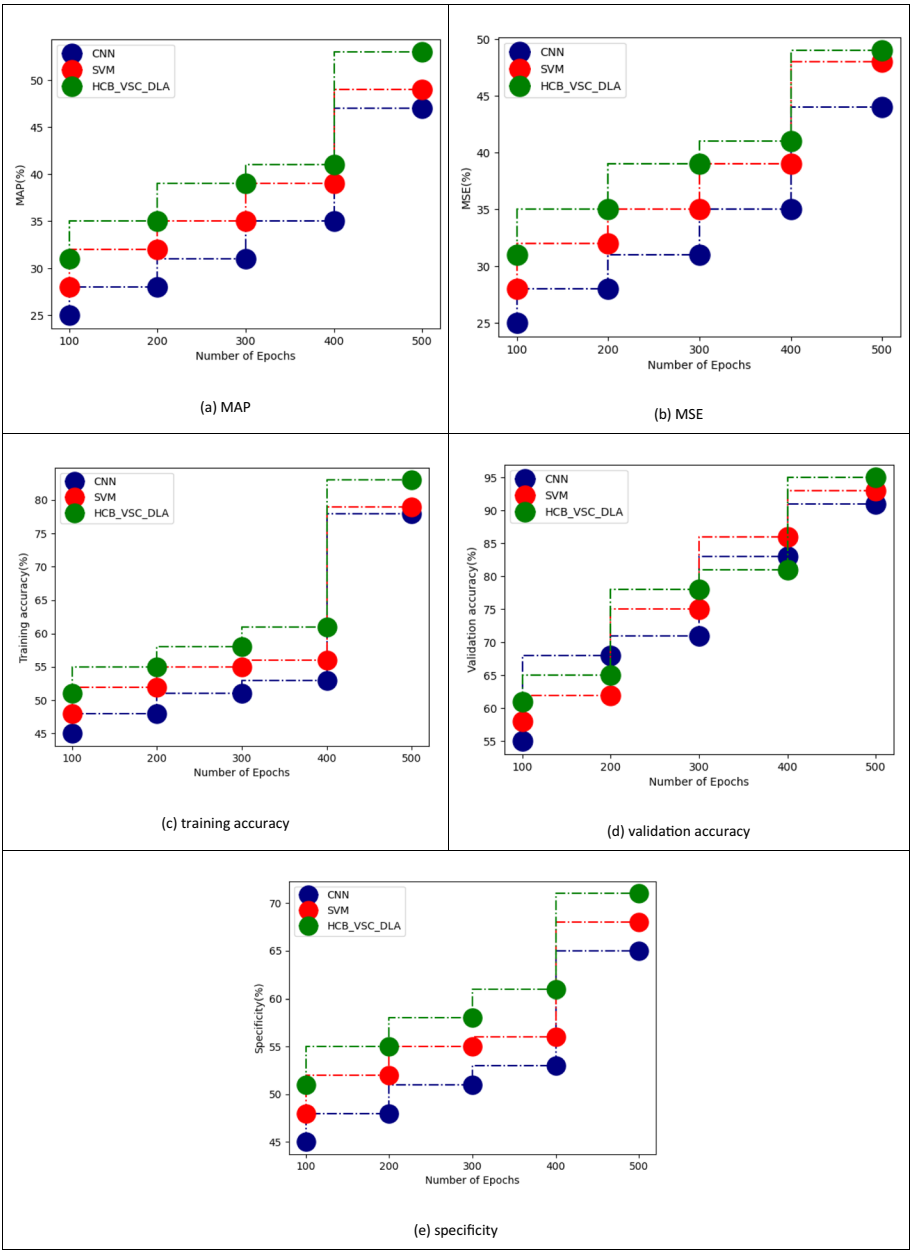

Fig. 7 Shanghai Tech A, B dataset analysis in terms of (**a**) specificity, (**b**) training accuracy, (**c**) validation accuracy, and (**e**) MAP

achieved by the SVM. Furthermore, the suggested method demonstrated remarkable outcomes on the CUHK dataset, exhibiting a MAP of 59%, an MSE of 61%, trained accuracy of 95%, validation precision of 95%, and specific of 88%. SVM attained a MAP of 55%, MSE of 58%, training precision of 89%, validation precision of 93%, and specificity of

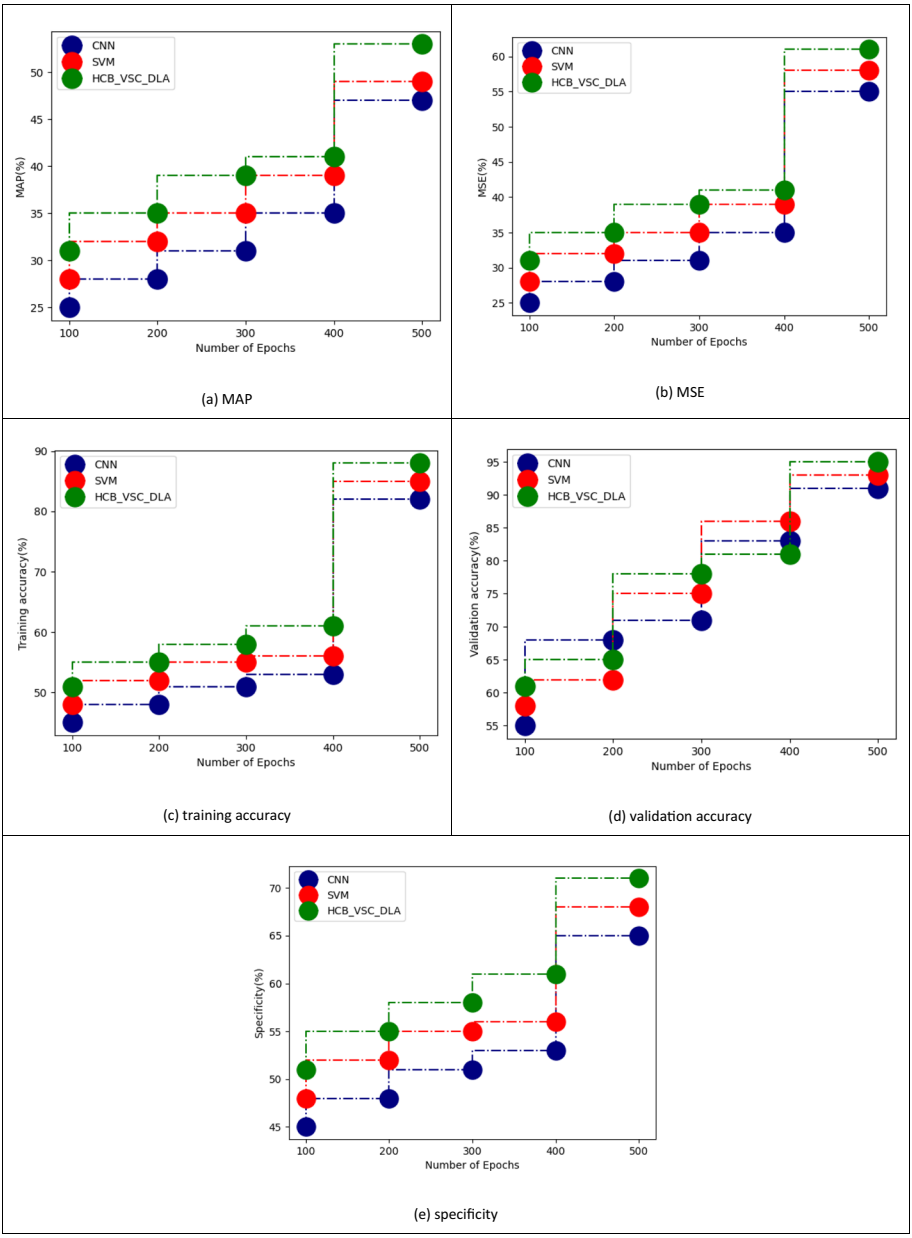

Fig. 8 Evaluation of the UCF-QNRF dataset with respect to (**a**) specificity, (**b**) training accuracy, (**c**) validation accuracy, and (**e**) MSE

86%, whereas the current CNN achieved a MAP of 52%, MSE of 55%, training precision of 84%, testing accuracy of 91%, and specificity of 82%. These results highlight the suggested technique's effectiveness and supremacy over current approaches on a variety of datasets about natural behavior in crowds.

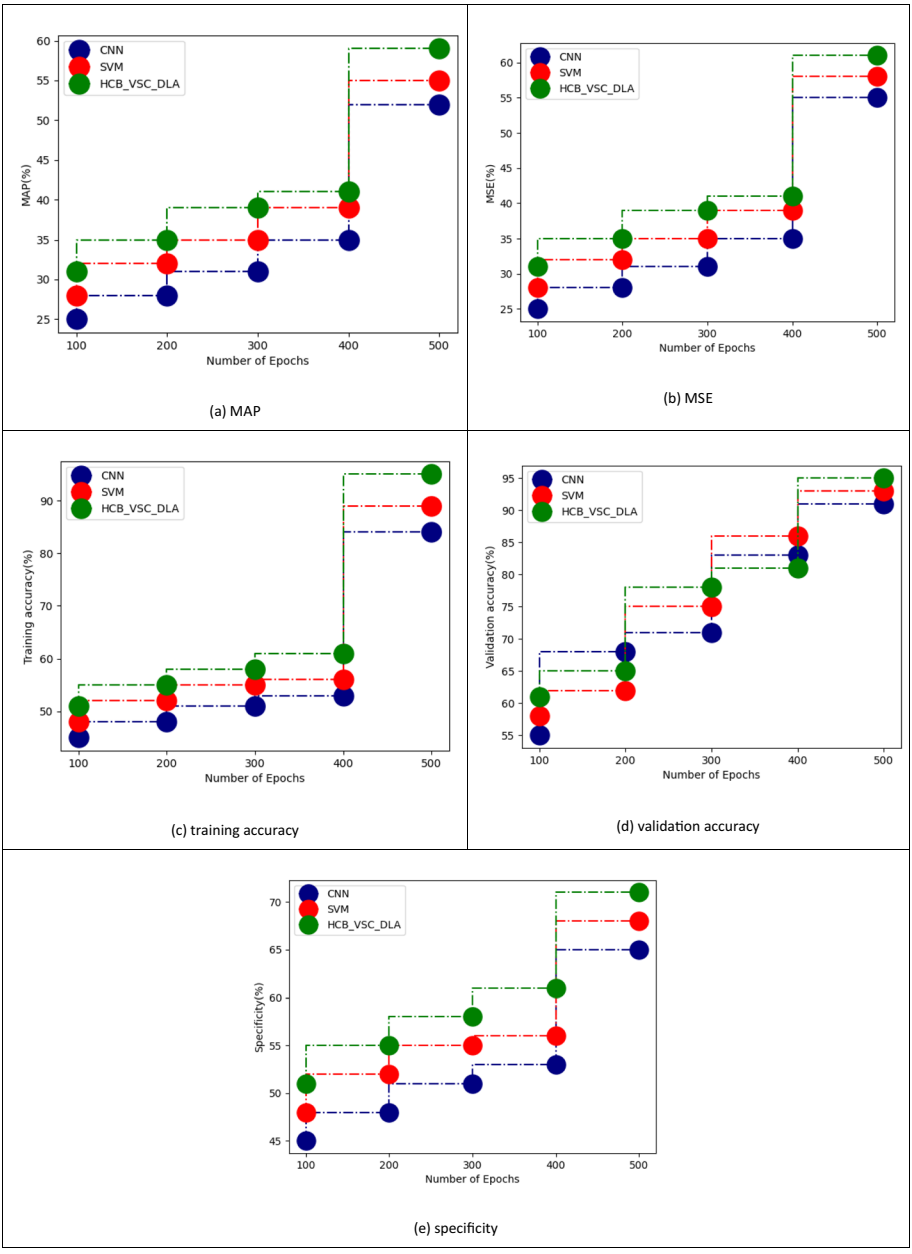

Fig. 9 Examination of the CUHK dataset concerning (**a**) particularity, (**b**) training precision, (**c**) validation accuracy, and (**e**) MAP

5 Conclusion

In this work, we used video segmentation and classification to build a novel approach for the investigation of human crowd behavior. By leveraging expectation–maximization-based ZFNet architecture for segmentation and transfer exponential Conjugate gradient neural networks for classification, we achieved promising results. Our experiments on real human activity databases demonstrated the superiority of our deep learning (DL) approach, with notable numerical findings including a MAP of 59%, MSE of 61%, and high training and validation accuracies of 95%, along with a specificity of 88%. Despite these advancements, limitations exist, notably the need for further optimization in control parameters and potential bias in segmentation networks when dealing with imbalanced data. Moving forward, future work will explore ensemble techniques and self-adaptive parameter control-based evolution for DL models, inspired by the success of our approach. Additionally, we aim to integrate multimodal data, such as audio or sensor information for depth and accuracy of crowd behaviour analysis.

Nomenclature *[d0,..., d9]*: Local Descriptor; *[D0..., Dt-1]*: SSIM value; *I*: Incomplete; *g*: Dimensional; *z*: Component memberships; $1, \pi_1, \ldots, \pi_k$: Mulr multinomial distribution; $Q\left(\Psi : \widehat{\Psi}^n\right)$: Conditional expectation; *v*: Observed data; z_i^n: Posterior probability; Γ_i: Kth component of the mixture; $\widehat{\psi}$: Parameter set; $\widehat{\Psi}(t+1)$: Fresh parameter; $\dot{\pi}_k^{n+1)}$: New estimates; z_n^n: Digamma function; $h \times w$: Height, width; *d*: Depth; x_{ij}: Input vector; f_{ks}: Function; y_{ij}: Vector output; $I_n(g)$: Average of a function g; $g\left(x_i\right)$: Independent random variables; W_n: Weight matrix; $B_{m(j)}$: Bias and m is the number of inputs; $\Delta(f*)$: Constant; $\inf_j \int_{\mathbb{R}^d}$: Fourier transform extension; *W*: Output weights; w_j^0: Random variables; $(a1, \ldots, am)$: Coefficients; *U(k)*: Projected convergence rate; *V(k)*: Observed consensus error; $z_i(k)$: Iterates; *z*: First term; *z**: Expected optimization error; $R(k)$: SGD's effectiveness; η_i: Rate of learning; d_i: Sample drawn; *d*: Random sample; Ω: Sample space; $h_{t+1} - h_t$: Random variable; w^t: The separation between the existing solution; w^*: Ideal solution; h_t: Random variable

Author contributions All authors are contributed equally to this work.

Funding No funding is involved in this work.

Data availability Data sharing not applicable to this article as no datasets were generated or analyzed during the current study.

Declarations

Ethics approval and consent to participate No participation of humans takes place in this implementation process.

Human and animal rights No violation of Human and Animal Rights is involved.

Conflict of interest Conflict of interest is not applicable in this work.

References

1. Tyagi B, Nigam S, Singh R (2022) A review of deep learning techniques for crowd behaviour analysis. Arch Computat Methods Eng 29(7):5427–5455
2. Chaudhary D, Kumar S, Dhaka VS (2022) Video based human crowd analysis using machine learning: a survey. Comput Methods Biomech Biomed Eng: Imaging Vis 10(2):113–131

3. Bruno A, Ferjani M, Sabeur Z, Arbab-Zavar B, Cetinkaya D, Johnstone L, ... Benaouda D (2022) High-level feature extraction for crowd behaviour analysis: a computer vision approach. In Image Analysis and Processing. ICIAP 2022 Workshops: ICIAP International Workshops, Lecce, Italy, May 23–27, 2022, Revised Selected Papers, Part II (pp. 59–70). Springer International Publishing, Cham
4. Kong YX, Wu RJ, Zhang YC, Shi GY (2023) Utilizing statistical physics and machine learning to discover collective behaviour on temporal social networks. Inf Process Manage 60(2):103190
5. Farooq MU, Mohamad Saad MN, Saleh Y, Daud Khan S (2022) Deep learning approach for divergence behaviour detection at high density crowd. In International Conference on Artificial Intelligence for Smart Community: AISC 2020, 17–18 December, UniversitiTeknologi Petronas, Malaysia (pp. 875–888). Springer Nature Singapore, Singapore
6. Sharma V, Mir RN, Singh C (2023) Scale-aware CNN for crowd density estimation and crowd behaviour analysis. Comput Electr Eng 106:108569
7. Bahamid A, Mohd Ibrahim A (2022) A review on crowd analysis of evacuation and abnormality detection based on machine learning systems. Neural Comput Appl 34(24):21641–21655
8. Bhuiyan MR, Abdullah J, Hashim N, Al Farid F (2022) Video analytics using deep learning for crowd analysis: a review. Multimed Tools Appl 81(19):27895–27922
9. Matkovic F, Ivasic-Kos M, Ribaric S (2022) A new approach to dominant motion pattern recognition at the macroscopic crowd level. Eng Appl Artif Intell 116:105387
10. Hou H, Wang L (2022) Measuring dynamics in evacuation behaviour with deep learning. Entropy 24(2):198
11. Pattan P, Arjunagi S (2022) A human behaviour analysis model to track object behaviour in surveillance videos. Measurement: Sensors 24:100454
12. Abpeikar S, Kasmarik K, Garratt M, Hunjet R, Khan MM, Qiu H (2022) Automatic collective motion tuning using actor-critic deep reinforcement learning. Swarm Evol Comput 72:101085
13. Zhang D, Li W, Gong J, Huang L, Zhang G, Shen S, ... Ma H (2022) HDRLM3D: a deep reinforcement learning-based model with human-like perceptron and policy for crowd evacuation in 3D environments. ISPRS Int J Geo-Inform 11(4):255
14. Lu Y, Ruan X, Huang J (2022) Deep reinforcement learning based on social spatial-temporal graph convolution network for crowd navigation. Machines 10(8):703
15. Liu T, Zheng Q, Tian L (2022) Application of distributed probability model in sports based on deep learning: deep belief network (DL-DBN) algorithm for human behaviour analysis. Comput Intell Neurosci 2022
16. Ha D, Tang Y (2022) Collective intelligence for deep learning: a survey of recent developments. Collective Intell 1(1):26339137221114870
17. Liang Z, Li L, Wang L (2022) Crowd-oriented behaviour simulation: reinforcement learning framework embedded with emotion model. In Artificial Intelligence: Second CAAI International Conference, CICAI 2022, Beijing, China, August 27–28, 2022, Revised Selected Papers, Part III (pp. 195–207). Springer Nature Switzerland, Cham
18. Choi T, Pyenson B, Liebig J, Pavlic TP (2022) Beyond tracking: using deep learning to discover novel interactions in biological swarms. Artif Life Robot 27(2):393–400
19. Poon KH, Wong PKY, Cheng JC (2022) Long-time gap crowd prediction using time series deep learning models with two-dimensional single attribute inputs. Adv Eng Inform 51:101482
20. Tiwari RG, Yadav SK, Misra A, Sharma A (2022) Classification of swarm collective motion using machine learning. In Human-Centric Smart Computing: Proceedings of ICHCSC 2022. Springer Nature Singapore, Singapore, pp 173–181
21 Chakole PD, Satpute VR, Cheggoju N (2022) Crowd behaviour anomaly detection using correlation of optical flow magnitude. J Phys: Conf Ser 2273(1):012023 (**IOP Publishing**)
22. Guo B, Liu Y, Liu S, Yu Z, Zhou X (2022) CrowdHMT: crowd intelligence with the deep fusion of human, machine, and IoT. IEEE Internet Things J 9(24):24822–24842
23. Tripathi SK (2022) Design and development of some methods and models for crowd analysis using computer vision and deep learning approaches.
24. Lalit R, Purwar RK (2022) Crowd abnormality detection using optical flow and glcm-based texture features. J Inform Technol Res (JITR) 15(1):1–15
25. Pai AK, Chandrahasan P, Raghavendra U, Karunakar AK (2023) Motion pattern-based crowd scene classification using histogram of angular deviations of trajectories. Vis Comput 39(2):557–567
26. Bala B, Kadurka RS, Negasa G (2022) Recognizing unusual activity with the deep learning perspective in crowd segment. In: A Fusion of Artificial Intelligence and Internet of Things for Emerging Cyber Systems. Springer, Cham, pp 171–181
27. Vidhyalakshmi M, Ramesh S, Bharathi ML, Kshirsagar PR, Rajaram A, Tirth V (2023) A comparative recognition research on excretory organism in medical applications using neural networks. Multimed Tools Appl 1–18

28. Shafiq M, Tian Z, Bashir AK, Du X, Guizani M (2020) CorrAUC: A malicious bot-IoT traffic detection method in IoT network using machine-learning techniques. IEEE Internet Things J 8(5):3242–3254
29. Shafiq M, Tian Z, Bashir AK, Du X, Guizani M (2020) IoT malicious traffic identification using wrapper-based feature selection mechanisms. Comput Secur 94:101863
30. Shafiq M, Tian Z, Bashir AK, Jolfaei A, Yu X (2020) Data mining and machine learning methods for sustainable smart cities traffic classification: a survey. Sustain Cities Soc 60:102177
31. Singh D, Kaur M, Alanazi JM, AlZubi AA, Lee HN (2022) Efficient evolving deep ensemble medical image captioning network. IEEE J Biomed Health Inform 27(2):1016–1025
32 Raina R, Gondhi NK, Chaahat, Singh D, Kaur M, Lee HN (2023) A systematic review on acute leukemia detection using deep learning techniques. Arch Computat Methods Eng 30(1):251–270

Authors and Affiliations

Shruti Garg[1] · Sudhir Sharma[2] · Sumit Dhariwal[3] · W. Deva Priya[4] · Mangal Singh[5] · S. Ramesh[6]

✉ S. Ramesh
rameshbe04@gmail.com

Shruti Garg
gshruti@bitmesra.ac.in

Sudhir Sharma
sudhir.sharma@jaipur.manipal.edu

Sumit Dhariwal
sumit.dhariwal@jaipur.manipal.edu

W. Deva Priya
w.devapriyya@gmail.com

Mangal Singh
mangal.etce@gmail.com

1 Department of CSE, Birla Institute of Technology, Mesra, Jharkhand 835215, India

2 Department of DSE, SIT, Manipal University Jaipur, Jaipur, India

3 Department of IT, SIT, Manipal University Jaipur, Jaipur, India

4 Department of Computer Science and Engineering, Saveetha School of Engineering, Saveetha Institute of Medical and Technical Sciences (SIMATS), Saveetha University, Chennai, India

5 Department of E&TC, Symbiosis Institute of Technology, Symbiosis International (Deemed University), Pune, Maharashtra 412115, India

6 Department of Networking and Communications, School of Computing, Faculty of Engineering and Technology, SRM Institute of Science and Technology Kattankulathur, Chennai, India