applied sciences

MDPI

Article

Deep Transfer Learning Enabled Intelligent Object Detection for Crowd Density Analysis on Video Surveillance Systems

Fadwa Alrowais [1], Saud S. Alotaibi [2], Fahd N. Al-Wesabi [3,*], Noha Negm [3,4], Rana Alabdan [5], Radwa Marzouk [6], Amal S. Mehanna [7] and Mesfer Al Duhayyim [8]

1 Department of Computer Sciences, College of Computer and Information Sciences, Princess Nourah Bint Abdulrahman University, Riyadh 11671, Saudi Arabia; faalrowais@pnu.edu.sa
2 Department of Information Systems, College of Computing and Information System, Umm Al-Qura University, Mecca 24382, Saudi Arabia; sotaibe@uqu.edu.sa
3 Department of Computer Science, College of Science & Art at Mahayil, King Khalid University, Abha 62529, Saudi Arabia; nbdelhamid@kku.edu.sa
4 Faculty of Science, Mathematics and Computer Science Department, Menoufia University, Shebeen El-Kom 32511, Egypt
5 Department of Information Systems, College of Computer and Information Science, Majmaah University, Al-Majmaah 11952, Saudi Arabia; r.alabdan@mu.edu.sa
6 Department of Information Systems, College of Computer and Information Sciences, Princess Nourah Bint Abdulrahman University, Riyadh 11671, Saudi Arabia; ramarzouk@pnu.edu.sa
7 Department of Digital Media, Faculty of Computers and Information Technology, Future University in Egypt, New Cairo 11845, Egypt; msamy@fue.edu.eg
8 Department of Computer Science, College of Sciences and Humanities-Aflaj, Prince Sattam bin Abdulaziz University, Al-Kharj 16278, Saudi Arabia; malduhayyim@psau.edu.sa
* Correspondence: falwesabi@kku.edu.sa

Citation: Alrowais, F.; Alotaibi, S.S.; Al-Wesabi, F.N.; Negm, N.; Alabdan, R.; Marzouk, R.; Mehanna, A.S.; Al Duhayyim, M. Deep Transfer Learning Enabled Intelligent Object Detection for Crowd Density Analysis on Video Surveillance Systems. *Appl. Sci.* **2022**, *12*, 6665. https://doi.org/10.3390/app12136665

Academic Editor: Giancarlo Mauri

Received: 10 May 2022
Accepted: 25 June 2022
Published: 30 June 2022

Abstract: Object detection is a computer vision based technique which is used to detect instances of semantic objects of a particular class in digital images and videos. Crowd density analysis is one of the commonly utilized applications of object detection. Since crowd density classification techniques face challenges like non-uniform density, occlusion, inter-scene, and intra-scene deviations, convolutional neural network (CNN) models are useful. This paper presents a Metaheuristics with Deep Transfer Learning Enabled Intelligent Crowd Density Detection and Classification (MDTL-ICDDC) model for video surveillance systems. The proposed MDTL-ICDDC technique mostly concentrates on the effective identification and classification of crowd density on video surveillance systems. In order to achieve this, the MDTL-ICDDC model primarily leverages a Salp Swarm Algorithm (SSA) with NASNetLarge model as a feature extraction in which the hyperparameter tuning process is performed by the SSA. Furthermore, a weighted extreme learning machine (WELM) method was utilized for crowd density and classification process. Finally, the krill swarm algorithm (KSA) is applied for an effective parameter optimization process and thereby improves the classification results. The experimental validation of the MDTL-ICDDC approach was carried out with a benchmark dataset, and the outcomes are examined under several aspects. The experimental values indicated that the MDTL-ICDDC system has accomplished enhanced performance over other models such as Gabor, BoW-SRP, Bow-LBP, GLCM-SVM, GoogleNet, and VGGNet.

Keywords: object detection; object tracking; video surveillance; computer vision; crowd density estimation; deep learning; parameter optimization

1. Introduction

Object detection is a computer technology related to computer vision and image processing that aims to determine and detect many target objects from still images or video data [1]. It widely comprises different important techniques, namely image processing,

pattern recognition, artificial intelligence (AI), and machine learning (ML). It finds applicability in different domains such as road traffic accident prevention, theft detection, traffic management, etc. [2,3]. Intelligent video surveillance is a vintage subject in the domain of image processing and computer vision that has recently become well known. It has numerous significant benefits, such as accurate data processing, low human resource cost, and effective information gathering organization [4]. Crowd density estimation is regarded as a significant application in visual surveillance, and it plays an important role in crowd management and monitoring. Specifically for service providers in public areas, a crowd density estimation system could show how many consumers are currently waiting and therefore provide an appropriate reference to send the bounded sources reasonably and effectively [5].

Crowd density estimation refers to the assessment of crowd dispersal and the particular number of people [6]. Crowd analysis has attracted substantial interest among researchers in recent years because of various reasons. The massive increase in the global population and in urbanization has resulted in an increase in such events as public demonstrations, sporting events, political rallies, and so on. Similar to other computer vision issues, crowd analysis faces numerous difficulties, such as inter-scene variations in appearance, occlusions, uneven distribution of people, high clutter, intra-scene and scale issues, non-uniform illumination, and an unclear viewpoint; these problems are immensely difficult to solve [7,8]. The perplexity of the issue along with the extensive array of applications for crowd analysis has led to an increased focus among research scholars in recent years.

Convolutional neural network (CNN) models [9,10] have reached successful outcomes in image processing and in the prediction of crowd density. Recent crowd density estimation methodologies are primarily dependent on regression or identification. Detection approaches can be implemented in cases that have a minimum number of persons and no occlusion-like detectors depending on closer frames [11,12]. Other methodologies that depend on regression are of two categories. The first includes those that identify handmade features in the image, namely texture feature and edge feature; next, regression function is selected for estimating aggregate person numbers [13]. Another one relies on deep neural networks and density map regression, and this technique is considered the best for estimating crowd density.

This paper presents a Metaheuristics with Deep Transfer Learning Enabled Intelligent Crowd Density Detection and Classification (MDTL-ICDDC) model on video surveillance systems. The proposed MDTL-ICDDC technique leverages a Salp Swarm Algorithm (SSA) with NASNetLarge model as a feature extraction in which the hyperparameter tuning process is performed by the SSA. Furthermore, a weighted extreme learning machine (WELM) technique was employed for crowd density and classification process. Finally, the krill swarm algorithm (KSA) is applied for effectual parameter optimization process and thereby improves the classification results. The experimental validation of the MDTL-ICDDC technique is carried out using benchmark dataset.

The rest of the paper is organized as follows. Section 2 offers a brief survey of recently developed crowd density estimation and classification models. Next, Section 3 provides the proposed MDTL-ICDDC technique for crowd classification on surveillance videos. Then, Section 4 validates the performance of the proposed model, and finally, Section 5 concludes with the major key findings of the work.

2. Related Works

This section presents a detailed literature review of the existing crowd density analysis models. Ding et al. [14] proposed a novel encoder-decoder CNN that combines the feature map in encoding and decoding subnetworks for estimating the number of people accurately. In addition, the authors present a new assessment methodology called the Patch Absolute Error (PAE) that is more applicable for measuring the accuracy of density maps. Zhu et al. [15] resolve crowd density evaluation problems for dense and sparse conditions. Consequently, it generates two contributions: (1) a network called Patch Scale Discriminant

Regression Network (PSDR). Considering an input crowd image, it splits the images as two patches and sends them into a regression network, which then yields a density map. It fuses the two patch density maps in order to predict the whole density map as the output. (2) A person classification activation map (CAM) technique is the other contribution.

In [16], the authors developed a Wi-Fi monitoring detection system that could capture smart phone passive Wi-Fi signal data involving a received signal strength indicator and MAC address. Next, the authors present a positioning model based on a dynamic fingerprint management strategy and smart phone passive Wi-Fi probe. In real time social activities, an individual might possess zero, one, two, or many smart phones with different Wi-Fi signals. Thus, it can be designed as a methodology for calculating the possibility of users generating one Wi-Fi signal to recognize the total population of people. Last, the authors developed a crowd density evaluation method based on a Wi-Fi packet positioning model.

In [17], the current research progression on density estimation and crowd counting has been comprehensively analyzed. First, the authors present the background of density estimation and crowd counting. Next, they summarized the traditional crowd counting method. Later, the authors focus on investigating the density estimation and crowd counting methodologies based on a CNN model. In [18], the authors proposed a crowd density estimation model, utilizing Hough circle transformation. Here, background and foreground datasets were segregated by ViBe technology and the segmentation of foreground datasets.

In Bouhlel et al. [19], a crowd density estimation method in an aerial image is proposed for examining a crowded region that shows an abnormal density. The presented technique comprises an inference and offline phase. The offline phase focused on generating a crowd model with a combination of handcrafted and relevant deep features designated by the use of the minimum-redundancy maximum-relevance (mRMR) method. During the inference phase, the previously generated models for classifying the aerial image patches yield the following four classifications: None, Sparse, Medium, and Dense. In [20], the network compression to the CNN-based crowd density estimation method is applied for reducing their computation and storage costs. In particular, the authors depend on l1-norm for selecting insignificant filters and physically pruning them. These models are trained to identify the insignificant filter and to increase the regression performance simultaneously.

The authors in [21] developed a new crowd density estimation model by the use of DL models for passenger flow recognition model in exhibition centers. At the initial stage, the difference amplitude feature and gray feature of the central pixel are derived to create the CLBP feature for obtaining more crowd-group description information. Moreover, the LR activation function is used for adding the non-linear factors to the CNN and exploiting dense blocks derived from crowd density estimation for calibrating the LR-CNN crowd density estimation model. Bhuiyan et al. [22] developed a fully convolutional neural network (FCNN) model for crowd density estimation on surveillance video captured by a camera at a distance. Li et al. [23] introduces a multi-scale feature fusion network (IA-MFFCN) depending upon reverse attention model that mapped the image into the crowd density map for counting purposes. Wang et al. [24] developed a lightweight CNN to estimate crowd density by the combination of the modified MobileNetv2 and the dilated convolution.

3. The Proposed Model

In this study, a new MDTL-ICDDC technique was established for effectual identification and classification of crowd density on a video surveillance system. The MDTL-ICDDC model initially presented an SSA with NASNetLarge model as a feature extraction model in which the hyperparameter tuning process is performed by the SSA. This was followed by the KSA-WELM model, which is employed for crowd-density and classification processes. Figure 1 illustrates the block diagram of MDTL-ICDDC technique.

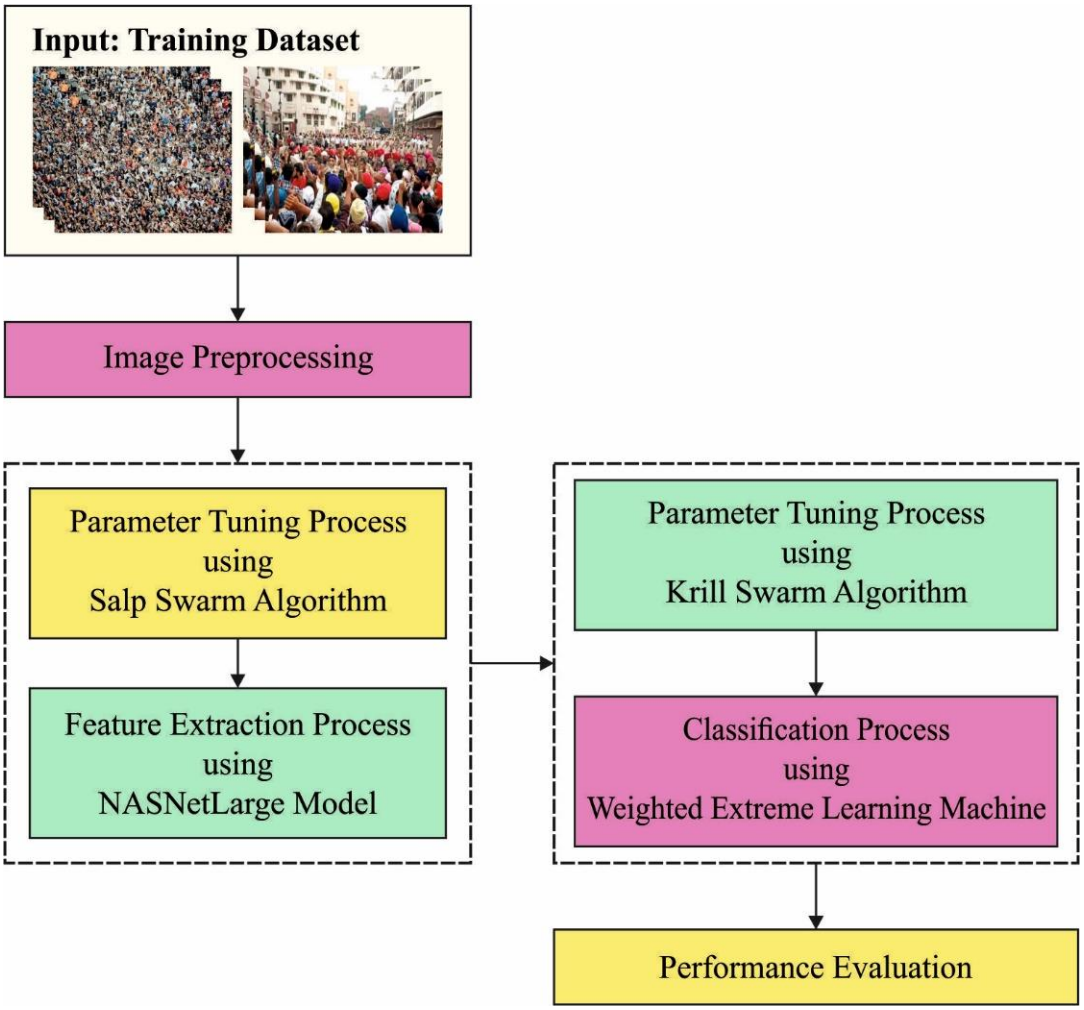

Figure 1. Block diagram of MDTL-ICDDC technique.

3.1. Feature Extraction Module

At the initial stage, the SSA with NASNetLarge model functions as a feature extraction model in which the hyperparameter tuning process is performed by the SSA. For determining the optimal convolution architecture for the dataset, a search algorithm can be used. Neural architecture search (NAS) is the most important search technique that the authors deployed in this network. Child network is shown to accomplish some accuracy on a validation set; in other words, it is used for convergence [25]. The subsequent accuracy value is utilized to upgrade the controller that consecutively generates better architecture over time. The policy gradient takes place to update the controller weight.

A new searching space was designed, which permits the better architecture found on the CIFAR-10 dataset (available at http://www.cs.toronto.edu/~kriz/cifar.html, accessed on 12 Febuary 2022) that generalized for large, high-resolution image datasets from the range of computation environments. To adapt input of depth of filtering and spatial dimension, this cell is sequentially stacked. In this technique, the convolution net overall architecture is predefined manually. They are composed of convolution cells that possess a similar shape as the original but are weighted in a different way. Two kinds of convolution cells have been taking place for rapidly developing scalable architecture for images of any size: (1) convolution cells return a feature map with a 2-fold reduction in width and height, and (2) convolution cells produce a feature map with the similar dimension. Figure 2 depicts the framework of NasNetLarge.

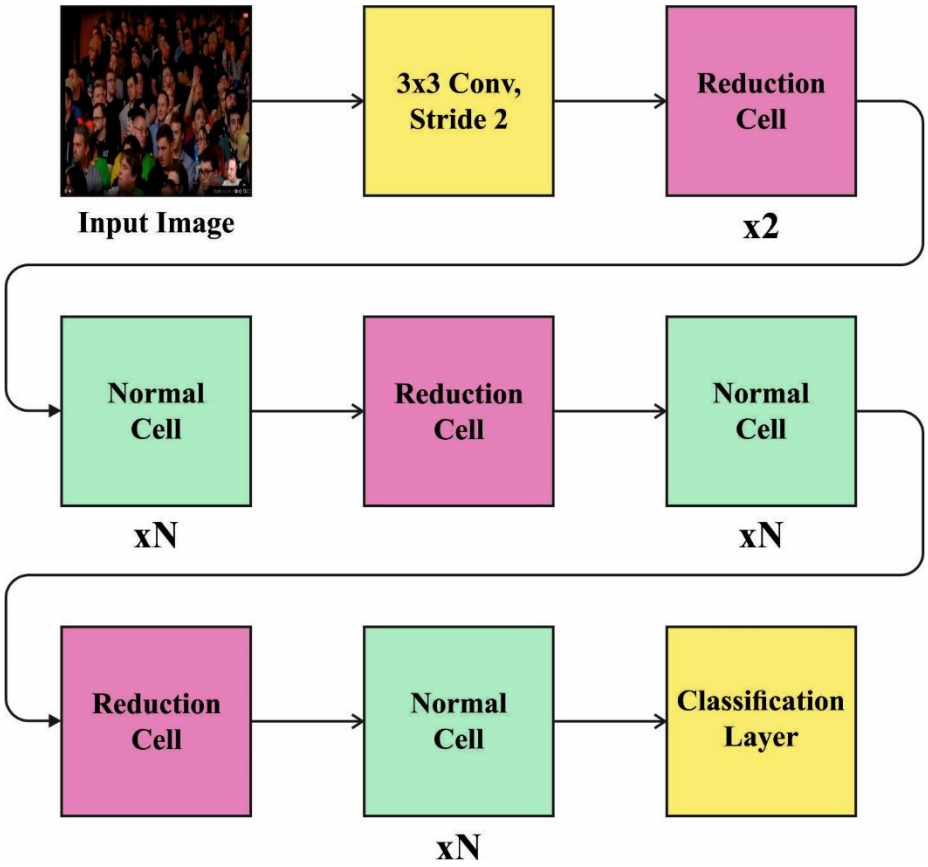

Figure 2. Architecture of NasNetLarge.

These two kinds of convolution cells are represented as Normal Cell and Reduction Cell, correspondingly. The primary process used for the cell's input gives a two-step stride to minimalize the cell's width and height. The convolution cells support striding because they consider each operation. The Normal as well as Reduction Cells architecture that the controller RNN search for dissimilar to convolution net. The searching region is utilized for searching each cell shape. There are 2 hidden states (HS), namely h (i) and h (i −1), presented for every cell in the searching space. Initially, the HS is the outcome of 2 cells in the prior 2 lower layers or input image, correspondingly. The controller RNN make recursive prediction on the remaining convolution cells on the basis of 2 primary HSs. The controller prediction for all the cells is ordered into B blocks, with every block comprising five prediction steps implemented by five discrete SoftMax classifications representing different selections of block elements.

Step 1. Choose an HS from h (i), h (i −1), or the set of formerly generate HS.

Step 2. In the similar option as in Step1, choose the next HS.

Step 3. In Step 1, select the HS that the authors need to employ.

Step 4. After, choose an HS in Step2, choose an operation to employ.

Step 5. Define how the output from Steps 3 & 4 would be integrated to make a novel HS. It can be helpful to apply the recently generated HS as an input for the subsequent block.

For the optimal hyperparameter tuning process, the SSA is utilized. It was initially established as a swarm intelligence process [26]; it can be stimulated by the foraging performance of the salp swarm that forms a chain. Similar to other algorithms, it finds an optimum solution via the cooperation and division of salps. The population is divided into the following categories: the leader and followers. Initially, the leaders lead the direction of the population, and next the follower follows them; consequently, the population forms a chain. The population X includes N agents are established as a matrix using D columns

and N rows. The target population from the searching region is a food source represented as F.

$$X = \begin{bmatrix} X_{1,1} & X_{1,2} & \cdots & X_{1,p} \\ X_{2,1} & X_{2,2} & \cdots & X_{2,p} \\ \vdots & \vdots & \vdots & \vdots \\ X_{N,1} & X_{N,2} & \cdots & X_{N,D} \end{bmatrix}, \quad (1)$$

whereas population size denotes N, D indicates dimension. The leader's location is rehabilitated.

$$X_{1,j} = \begin{cases} F_j + c_1 \times \left(\left(ub_j - 1b_j \right) \times c_2 + 1b_j \right) c_3 \geq 0.5 \\ F_j - c_1 \times \left(\left(ub_j - 1b_j \right) \times c_2 + 1b_j \right) c_3 \leq 0.5 \end{cases} \quad (2)$$

Now $X_{1,j}$ and F_j mean the jth parameter of the leader location and food source, respectively. c_1 signifies a control variable adoptively decreased through the iteration and calculated. The role of exploration and exploitation is to determine SSA. c_2 and c_3 are created within $[0, 1]$. ub_j and lb_j indicates the jth variable of upper and lower limits, respectively.

$$c_1 = 2 \times e^{-\left(\frac{4 \times l}{L}\right)^2}. \quad (3)$$

Here, l and L denote the existing and maximal iterations, respectively. The follower location is rehabilitated as [27].

$$X_{ij} = \frac{1}{2} \times \left(X_{ij} + X_{i-1j} \right), \quad (4)$$

where $i \geq 2$ and X_{ij} indicates the jth parameter in the position of i-th follower.

3.2. Crowd Density Classification Module

Once the feature vectors are created, the next stage is to classify the crowd density with the WELM model. WELM is an enhanced version of ELM that manages imbalanced class distribution data [28]. The WELM approach presented the weighted matrix W similar to how the original ELM model works to balance the data distribution that weakens the majority class and strengthens the minority class. The W represents a matrix of misclassification values according to class distribution. The W in (5) signifies diagonal matrixes with $N \times N$ dimensional in which N indicates the overall dataset, and $\#(t_i)$ denotes the overall amount of samples that belongings to class t_i.

$$W_{ii} = \frac{1}{\#(t_i)} \quad (5)$$

The WELM is based on the standardized ELM that minimizes the error vector ξ, as well as minimizes the output weight norm β to have improved generalization outcomes. In the following, the mathematical expression of the minimization problem is given.

$$\begin{aligned} &minimize : L_{P_{EM}} = \tfrac{1}{2}\|\beta\|^2 + \tfrac{1}{2} CW \sum_{i=1}^{N} \|\xi_i\|^2 \\ &subject\ to : h(x)\beta = t_i^T - \xi_i^T, i = 1, \ldots, N \end{aligned} \quad (6)$$

The solution of β is obtained from (6) on the basis of KKT condition into (7) if N is smaller and (8) if N is larger.

$$\beta = H^T \left(\frac{I}{C} + WHH^T \right)^{-1} WT \quad (7)$$

$$\beta = \left(-\frac{I}{C} + WHH^T \right)^{-1} H^T WT \quad (8)$$

For multi-class classification with m class, all the labels are mapped into a vector of $[-1, 1]$ through length of m; for example, a dataset that is categorized into the 2nd classes from three classes are $[-1, 1, -1]$. The output $f(x) = HB$ in multi-class classification is a vector $f(x) = [f_1(x), f_2(x), f_m(x)]$ and the class label is evaluated by the following equation. Figure 3 showcases the framework of WELM.

$$label\ of\ x = \arg max(x) \tag{9}$$

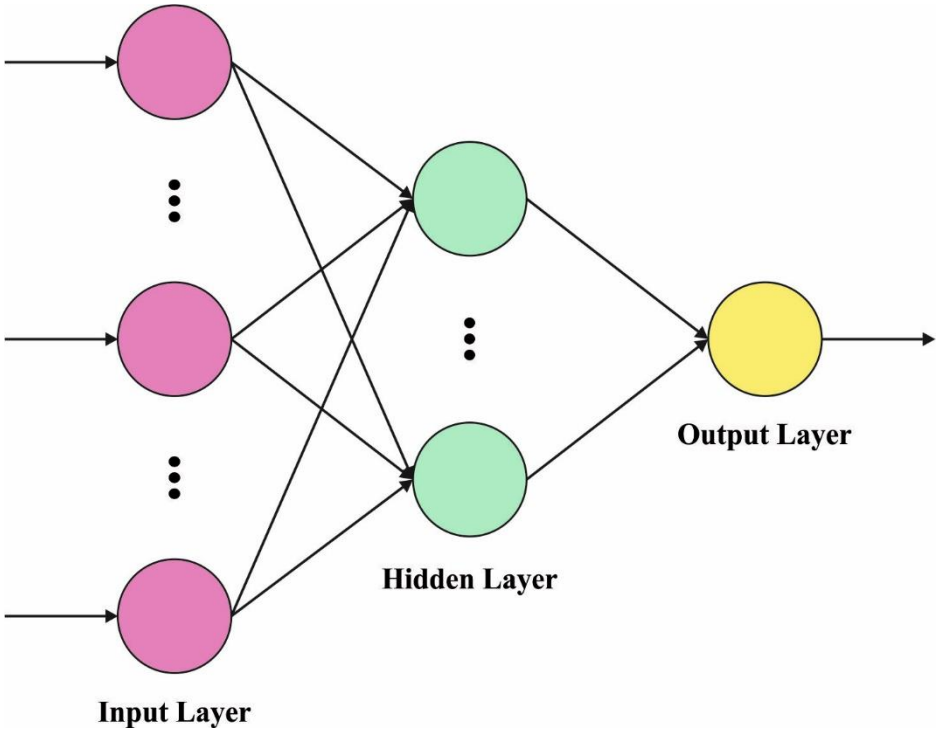

Figure 3. Structure of Weighted ELM.

3.3. Parameter Tuning Process

For the optimal adjustment of the WELM parameters, the KSA is utilized in this study. The krill swarm algorithm (KSA) [29] is a commonly used intelligent optimization technique, due to its benefits of strong search diversity, few adjusted parameters, and simple operation. The KSA approach originated from mutual communication and krill foraging. In the presented approach, the location of every individual krill represents a potential solution. The location of each krill is commonly specified in the following:

1. Individual swimming due to population migration:

$$N_i^{new} = N^{\max}\alpha_i + \omega_n N_i^{old} \tag{10}$$

$$\alpha_i = \alpha_i^{loca1} + \alpha_i^{target} \tag{11}$$

From the equation, the maximum induction speed can be represented as $N^{\max}$, and taken as 0.01 $\left(\mathrm{ms}^{-1}\right)$, and ω_n refers to the inertia weight of motion-induced range from [0, 1]. N_i^{old} represents prior movement, whereas α_i^{target} and α_i^{local} indicate the target and the existing positions, respectively.

2. Foraging behaviour:

$$F_i = V_f\beta_i + \omega_f F_i^{old} \tag{12}$$

$$\beta_i = \beta_i^{food} + \beta_i^{best} \tag{13}$$

Now, the individual direction of foraging for krill was denoted as β_i. β_i indicates the attractive direction of food, and β_i^{best} indicates the direction of individual krill with the optimal fitness value. V_f represents the foraging speed that takes as 0.02 $\left(\mathrm{ms}^{-1}\right)$, and ω_f represents the inertia weight that ranges from [0, 1]. F_i^{old} signifies the location change due to the preceding foraging motion of the *i-th* individual krill; F_i denotes the location change due to the existing foraging motion of the individual *i-th* krill.

3. Random diffusion of individual krill:

$$D_i = D^{\max}\delta \tag{14}$$

In Equation (14), $D^{\max}$ indicates the maximal disturbance (diffusion) velocity, and δ denotes an arbitrary direction vector that ranges from $[-1,\ 1]$; D_i signifies the location change due to arbitrary diffusion of *i*-th individual krill.

The swimming direction of each krill can be defined by the fusion of abovementioned factors that change in the direction with minimum fitness value. The foraging and induced motions have local and global searching functions. After the process is iteratively updated, two measures are simultaneously implemented, which make stable and powerful optimization algorithms.

The location vector from t to $t + \Delta t$ is formulated by:

$$X_i(t + \Delta t) = X_t(t) + \Delta t \frac{dX_i}{dt} \tag{15}$$

Here, Δt indicates the factor for the step size.

$$\Delta t = C_t \sum_{j=1}^{NV} (UB_j - LB_j) \tag{16}$$

Now, NV indicates the overall amount of parameters, whereas LB_j and UB_j denote the upper as well as lower bounds of the *j*-th parameter.

The KSA system grows a fitness function (FF) for attaining higher classifier performance. It resolves the positive integer for denoting the best efficiency of candidate results. In this work, the minimization of the classification error rate is measured FF, as shown in Equation (17).

$$\begin{aligned} fitness(x_i) &= ClassifierErrorRate(x_i) \\ &= \frac{number\ of\ misclassified\ samples}{Total\ number\ of\ samples} * 100 \end{aligned} \tag{17}$$

4. Results and Discussion

In this section, the experimental validation of the MDTL-ICDDC model is tested with a dataset comprising 1000 images under four class labels. The MDTL-ICDDC model is simulated using Python 3.6.5 tool on a PC i5-8600k, GeForce 1050Ti 4 GB, 16 GB RAM, 250GB SSD, and 1 TB HDD. The parameter settings are given as follows: learning rate: 0.01, dropout: 0.5, batch size: 5, epoch count: 50, and activation: ReLU. Few sample images are demonstrated in Figure 4. The details related to the dataset are given in Table 1.

Table 1. Dataset details.

Labels	Class Name	No. of Instances
Class-0	Dense Crowd	250
Class-1	Medium Dense Crowd	250
Class-2	Sparse Crowd	250
Class-3	No Crowd	250
Total No. of Instances		1000

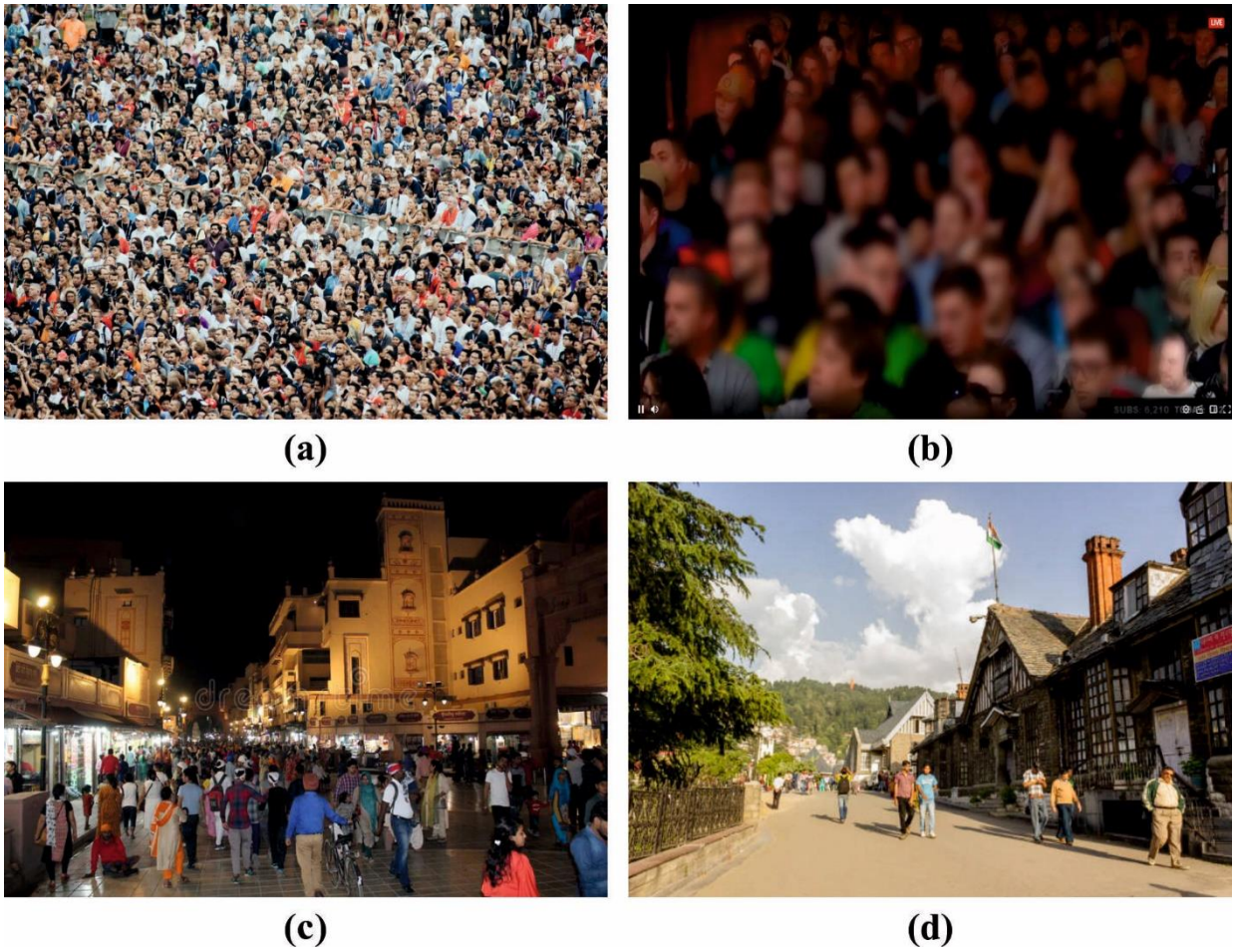

Figure 4. Sample Crowd Density Images. (**a**) Dense Crowd, (**b**) Medium Dense Crowd, (**c**) Sparse Crowd, (**d**) No Crowd.

Figure 5 demonstrates a set of confusion matrices formed by the MDTL-ICDDC model on distinct epoch counts. On epoch 200, the MDTL-ICDDC model has recognized 223, 220, 236, and 243 images under classes 0–3 respectively. Moreover, on epoch 600, the MDTL-ICDDC technique has recognized 206, 203, 234, and 241 images under classes 0–3 respectively. At the same time, on epoch 1000, the MDTL-ICDDC approach has recognized 197, 178, 233, and 234 images under classes 0–3 respectively. In line with, epoch 1200, the MDTL-ICDDC system has recognized 227, 178, 236, and 242 images under classes 0–3 respectively.

Table 2 and Figure 6 offer a detailed crowd density classification outcome of the MDTL-ICDDC model under distinct epochs and classes. The results indicated that the MDTL-ICDDC model has gained effectual outcomes under all classes and epochs. For instance, with 200 epochs, the MDTL-ICDDC model has provided average $accu_y$, $prec_n$, $reca_l$, F_{score}, and $G_{measure}$ of 96.10%, 92.20%, 92.20%, 92.18%, and 92.19% respectively. Likewise, with 600 epochs, the MDTL-ICDDC technique has obtainable average $accu_y$, $prec_n$, $reca_l$, F_{score}, and $G_{measure}$ of 94.20%, 88.38%, 88.40%, 88.31%, and 88.35% respectively. Similarly, with 1000 epochs, the MDTL-ICDDC system has provided average $accu_y$, $prec_n$, $reca_l$, F_{score}, and $G_{measure}$ of 92.10%, 84.06%, 84.20%, 83.94%, and 84.04% respectively. Eventually, with 1200 epochs, the MDTL-ICDDC methodology has offered average $accu_y$, $prec_n$, $reca_l$, F_{score}, and $G_{measure}$ of 96.40%, 92.82%, 92.80%, 92.80%, and 92.80% respectively.

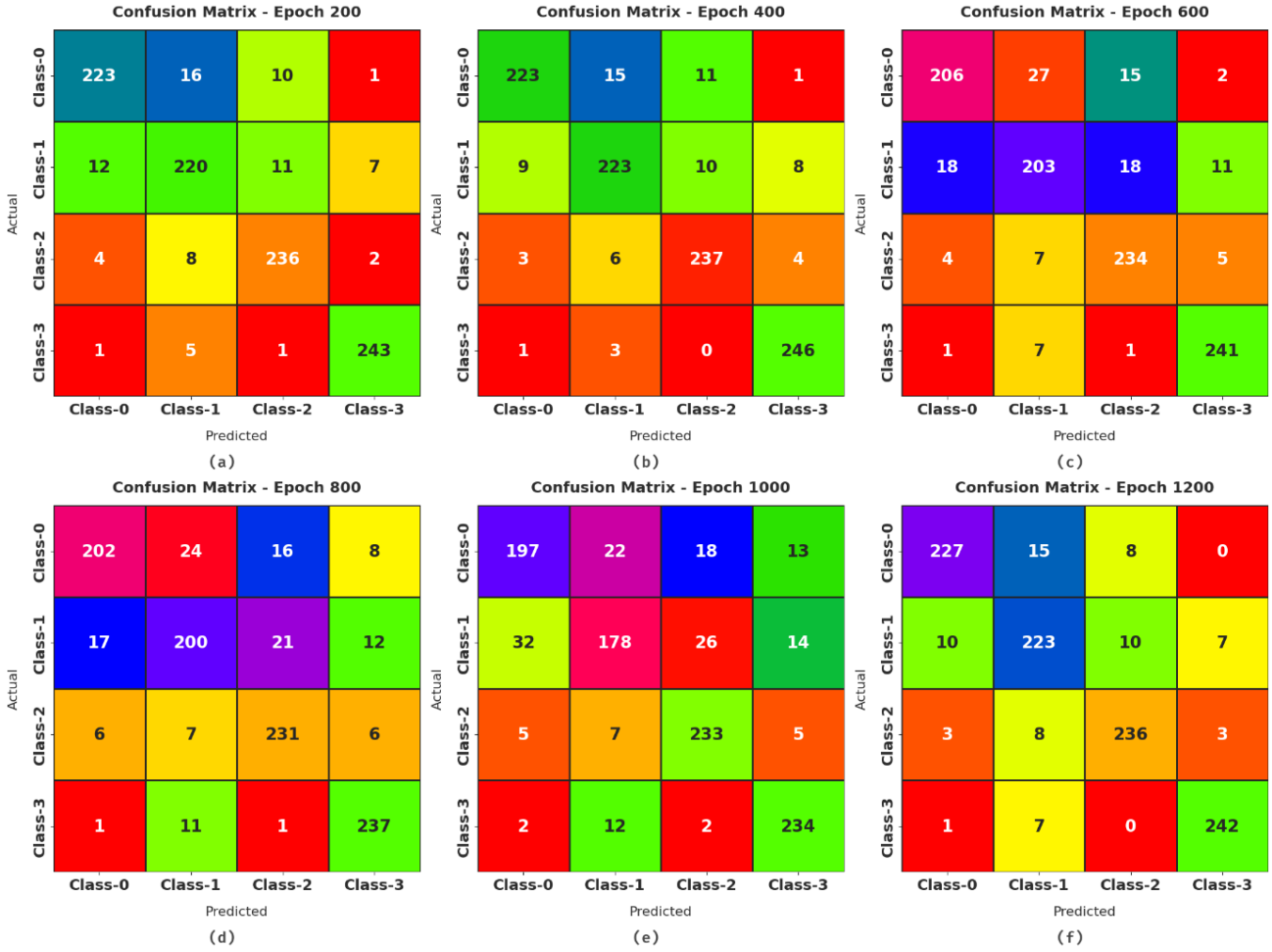

Figure 5. Confusion matrices of MDTL-ICDDC technique (**a**) epoch 200, (**b**) 400 epoch, (**c**) epoch 600, (**d**) epoch 800, (**e**) epoch 1000, and (**f**) epoch 1200.

The training accuracy (TA) and validation accuracy (VA) attained by the MDTL-ICDDC method on test dataset are demonstrated in Figure 7. The experimental outcome implied that the MDTL-ICDDC model has gained maximal values of TA and VA. Specfically, the VA seemed superior to the TA.

The training loss (TL) and validation loss (VL) achieved by the MDTL-ICDDC model on test dataset are displayed in Figure 8. The experimental outcome exposed that the MDTL-ICDDC technique has been able least values of TL and VL. Specifically, the VL seemed lower than the TL.

A brief precision-recall examination of the MDTL-ICDDC method on test dataset is portrayed in Figure 9. An observation of the figure shows that the MDTL-ICDDC system has been able to achieve maximal precision-recall performance under all classes.

A detailed ROC investigation of the MDTL-ICDDC approach on test dataset is depicted in Figure 10. The results indicated that the MDTL-ICDDC model has exhibited its ability in categorizing four different classes 0–3 on the test dataset.

Table 3 and Figure 11 inspect a comparative $prec_n$ inspection of the MDTL-ICDDC model with other models under distinct classes [30,31]. The experimental results indicated that the Gabor and BoW-SRP models have shown lower classification results with least average $prec_n$ of 61.83% and 68.33% respectively. Likewise, the BoW-LBP and GLCM-SVM models have accomplished slightly improved average $prec_n$ values of 74.68% and 75.47% respectively. At the same time, the GoogleNet and VGGNet techniques have resulted in

reasonable average $prec_n$ values of 82.98% and 86.14% respectively. However, the MDTL-ICDDC model has gained maximum average $prec_n$ of 92.90%.

Table 2. Result analysis of MDTL-ICDDC technique with distinct measures and epochs.

Class Labels	Accuracy	Precision	Recall	F-Score	G-Measure
Epoch-200					
Class-0	95.60	92.92	89.20	91.02	91.04
Class-1	94.10	88.35	88.00	88.18	88.18
Class-2	96.40	91.47	94.40	92.91	92.92
Class-3	98.30	96.05	97.20	96.62	96.62
Average	96.10	92.20	92.20	92.18	92.19
Epoch-400					
Class-0	96.00	94.49	89.20	91.77	91.81
Class-1	94.90	90.28	89.20	89.74	89.74
Class-2	96.60	91.86	94.80	93.31	93.32
Class-3	98.30	94.98	98.40	96.66	96.68
Average	96.45	92.90	92.90	92.87	92.89
Epoch-600					
Class-0	93.30	89.96	82.40	86.01	86.10
Class-1	91.20	83.20	81.20	82.19	82.19
Class-2	95.00	87.31	93.60	90.35	90.40
Class-3	97.30	93.05	96.40	94.70	94.71
Average	94.20	88.38	88.40	88.31	88.35
Epoch-800					
Class-0	92.80	89.38	80.80	84.87	84.98
Class-1	90.80	82.64	80.00	81.30	81.31
Class-2	94.30	85.87	92.40	89.02	89.08
Class-3	96.10	90.11	94.80	92.40	92.43
Average	93.50	87.00	87.00	86.90	86.95
Epoch-1000					
Class-0	90.80	83.47	78.80	81.07	81.10
Class-1	88.70	81.28	71.20	75.91	76.07
Class-2	93.70	83.51	93.20	88.09	88.22
Class-3	95.20	87.97	93.60	90.70	90.74
Average	92.10	84.06	84.20	83.94	84.04
Epoch-1200					
Class-0	96.30	94.19	90.80	92.46	92.48
Class-1	94.30	88.14	89.20	88.67	88.67
Class-2	96.80	92.91	94.40	93.65	93.65
Class-3	98.20	96.03	96.80	96.41	96.42
Average	96.40	92.82	92.80	92.80	92.80

Figure 6. Average analysis of MDTL-ICDDC technique (**a**) epoch 200, (**b**) 400 epoch, (**c**) epoch 600, (**d**) epoch 800, (**e**) epoch 1000, and (**f**) epoch 1200.

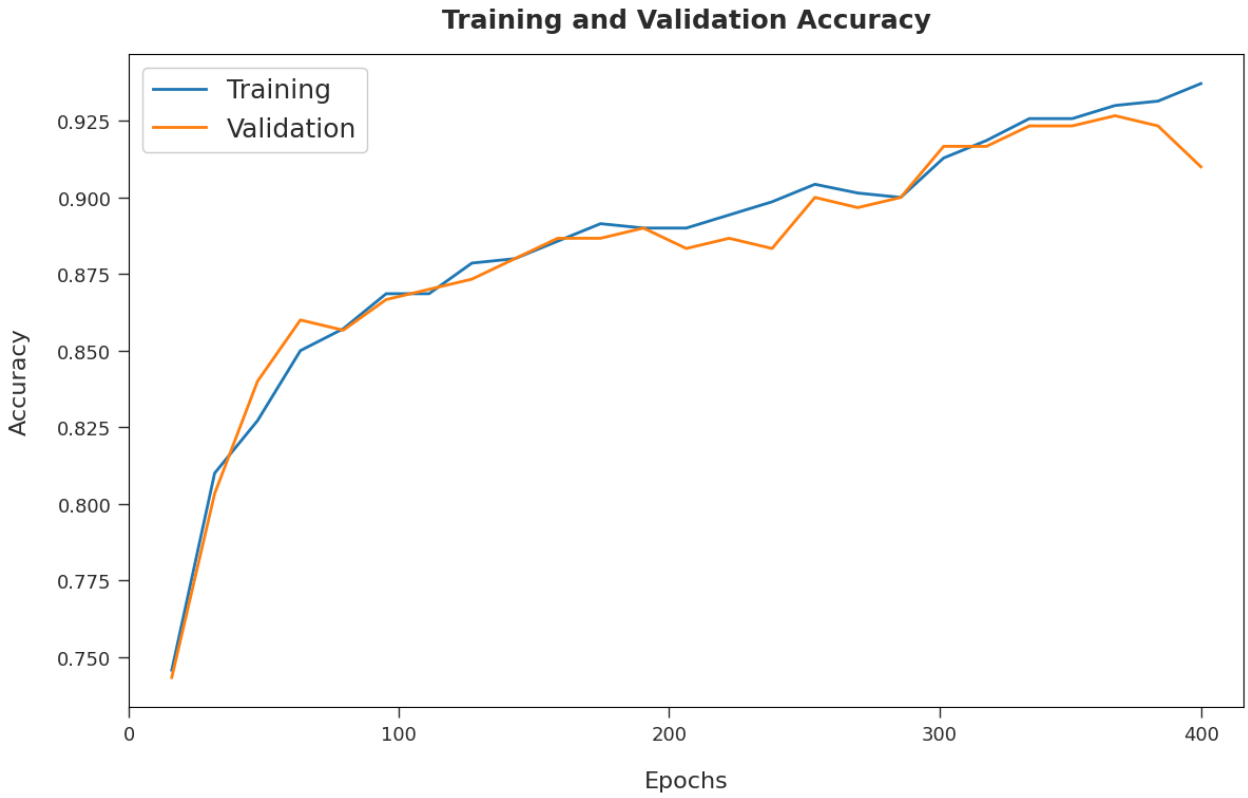

Figure 7. TA and VA analysis of MDTL-ICDDC technique.

Figure 8. TL and VL analysis of MDTL-ICDDC technique.

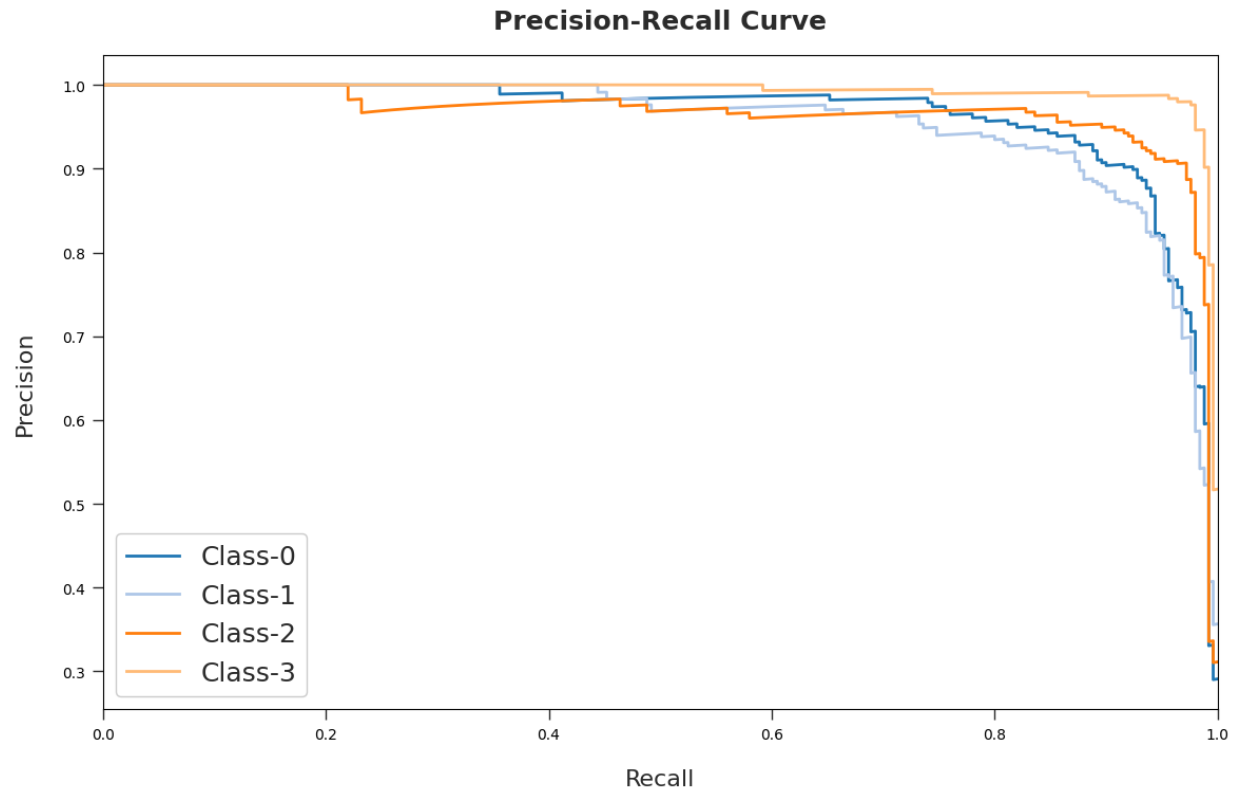

Figure 9. Precision-recall curve analysis of MDTL-ICDDC technique.

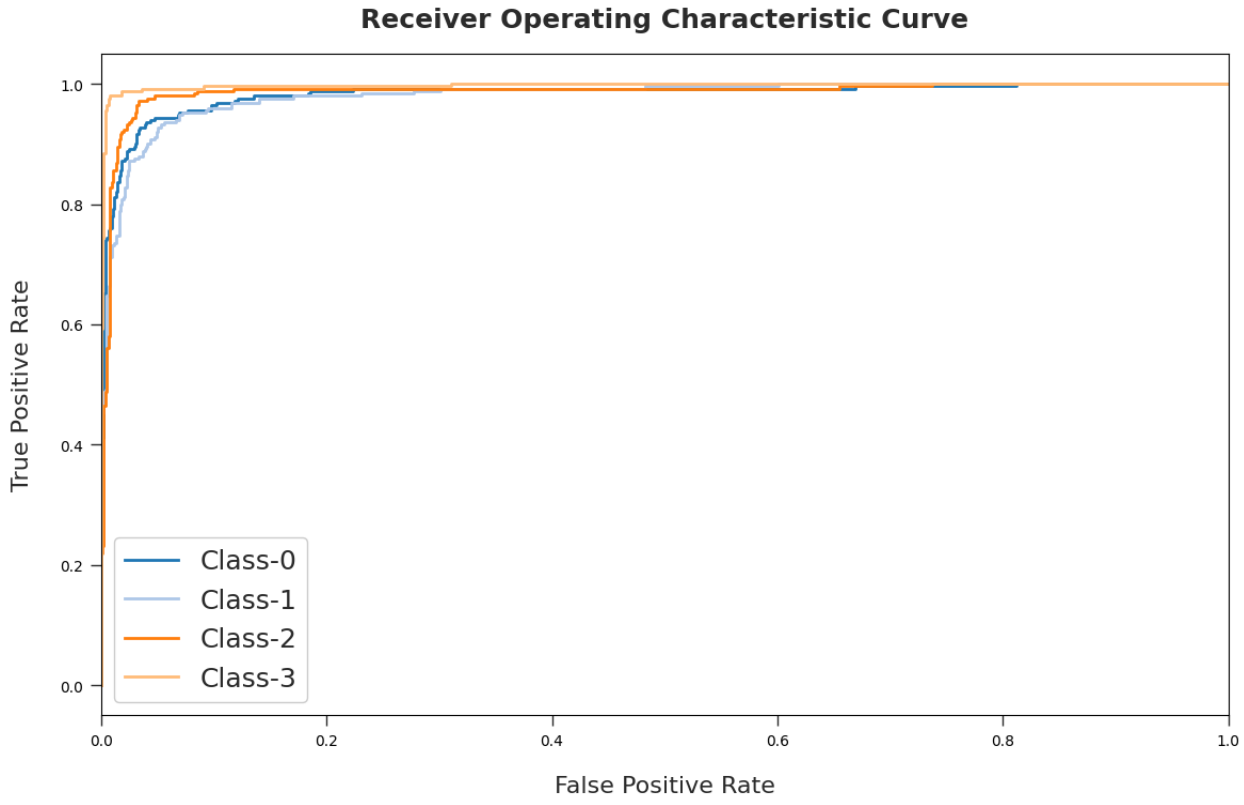

Figure 10. ROC curve analysis of MDTL-ICDDC technique.

Table 3. Precision analysis of MDTL-ICDDC technique with existing algorithms under various classes.

	Precision (%)				
Methods	**Class-0**	**Class-1**	**Class-2**	**Class-3**	**Average**
Gabor	57.80	47.80	56.50	85.20	61.83
BoW-SRP	76.00	49.60	59.10	88.60	68.33
Bow-LBP	75.80	55.80	72.20	94.90	74.68
GLCM-SVM	72.70	70.33	78.42	80.41	75.47
GoogleNet	74.05	89.38	78.48	90.01	82.98
VGGNet	79.42	82.11	89.78	93.23	86.14
MDTL-ICDDC	94.49	90.28	91.86	94.98	92.90

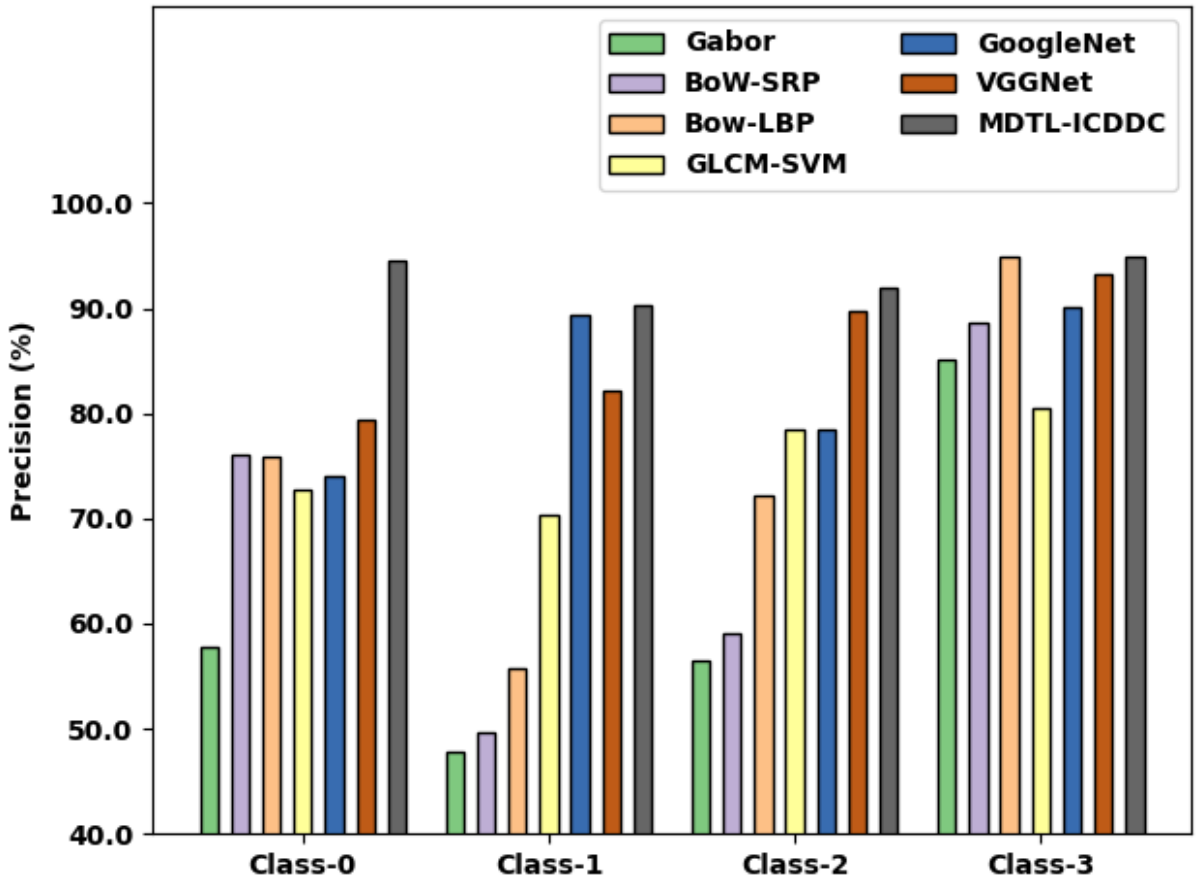

Figure 11. Precision analysis of MDTL-ICDDC technique under various classes.

Table 4 and Figure 12 demonstrate a comparative $reca_l$ analysis of the MDTL-ICDDC approach with other models under distinct classes. The experimental results indicated that the Gabor and BoW-SRP models have shown lower classification results with least average $reca_l$ of 62.30% and 67.85% respectively. Moreover, the BoW-LBP and GLCM-SVM models have accomplished somewhat enhanced average $reca_l$ values of 74.15% and 73.52% correspondingly. Simultaneously, the GoogleNet and VGGNet techniques have resulted in reasonable average $reca_l$ values of 85.26% and 82.78% respectively. But, the MDTL-ICDDC model has gained maximal average $reca_l$ of 92.90%.

Table 4. Recall analysis of MDTL-ICDDC technique with existing algorithms under various classes.

	Recall (%)				
Methods	**Class-0**	**Class-1**	**Class-2**	**Class-3**	**Average**
Gabor	52.00	47.40	58.80	91.00	62.30
BoW-SRP	63.40	49.60	67.00	91.40	67.85
Bow-LBP	74.00	61.80	67.00	93.80	74.15
GLCM-SVM	73.59	69.57	79.17	71.74	73.52
GoogleNet	79.60	83.99	87.18	90.26	85.26
VGGNet	76.08	82.46	89.82	82.75	82.78
MDTL-ICDDC	89.20	89.20	94.80	98.40	92.90

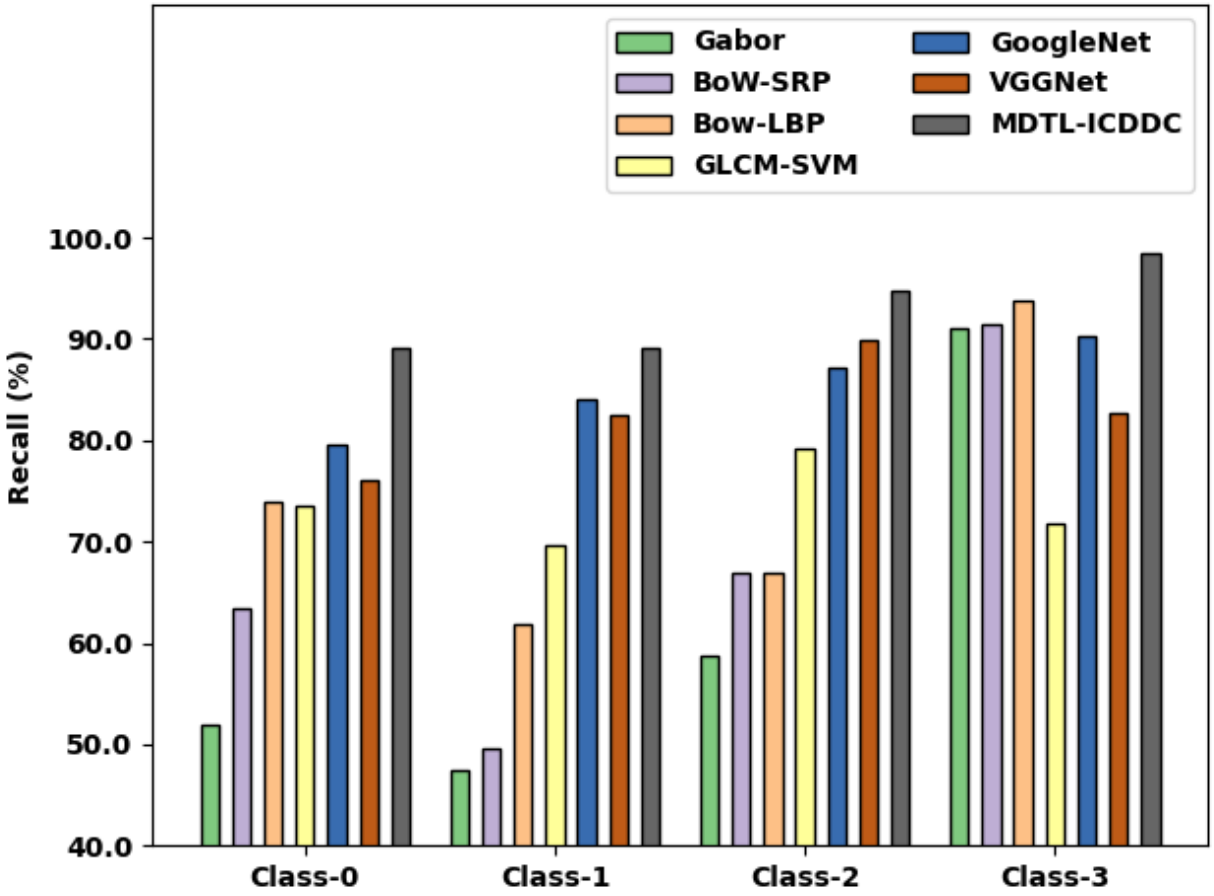

Figure 12. Recall analysis of MDTL-ICDDC technique under various classes.

Table 5 and Figure 13 depict a comparative $accu_y$ examination of the MDTL-ICDDC model with other algorithms under distinct classes. The experimental results indicated that the Gabor and BoW-SRP approaches have shown lower classification results with least average $accu_y$ of 71.83% and 80.40% respectively. Alongside these results, the BoW-LBP and GLCM-SVM methods have accomplished slightly improved average $accu_y$ values of 84.04% and 79.78% respectively. These results are followed by the GoogleNet and VGGNet approaches, which have resulted in reasonable average $accu_y$ values of 84.40 and 84.75% respectively. At last, the MDTL-ICDDC method has gained higher average $accu_y$ of 96.45%.

Table 5. Accuracy analysis of MDTL-ICDDC technique with existing algorithms under various classes.

	Accuracy (%)				
Methods	**Class-0**	**Class-1**	**Class-2**	**Class-3**	**Average**
Gabor	55.13	80.83	65.67	85.67	71.83
BoW-SRP	79.65	86.95	70.86	84.15	80.40
Bow-LBP	91.89	80.86	74.86	88.55	84.04
GLCM-SVM	72.43	82.21	70.99	93.48	79.78
GoogleNet	93.00	73.23	92.88	78.50	84.40
VGGNet	86.73	82.73	78.95	90.57	84.75
MDTL-ICDDC	96.00	94.90	96.60	98.30	96.45

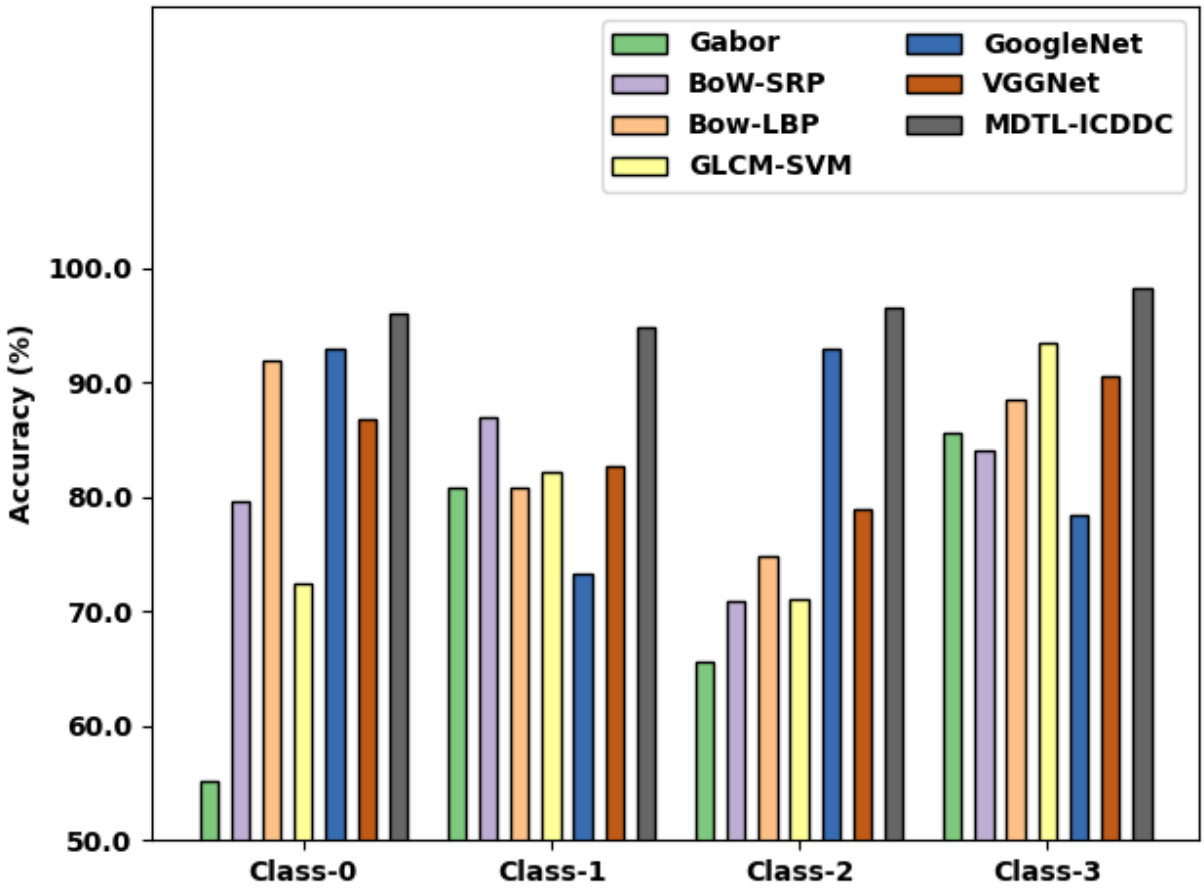

Figure 13. Accuracy analysis of MDTL-ICDDC technique under various classes.

Table 6 and Figure 14 illustrate a comparative F_{score} inspection of the MDTL-ICDDC algorithm with other techniques under distinct classes. The experimental results indicated that the Gabor and BoW-SRP models have exposed lesser classification results with minimal average F_{score} of 61.98% and 67.88% respectively. In addition, the BoW-LBP and GLCM-SVM models have accomplished somewhat higher average F_{score} values of 74.35% and 87.99% respectively.

Table 6. F-score analysis of MDTL-ICDDC technique with existing algorithms under various classes.

	F-Score (%)				
Methods	**Class-0**	**Class-1**	**Class-2**	**Class-3**	**Average**
Gabor	54.70	47.60	57.60	88.00	61.98
BoW-SRP	69.10	49.60	62.80	90.00	67.88
Bow-LBP	74.90	58.60	69.50	94.40	74.35
GLCM-SVM	89.44	85.78	91.64	85.09	87.99
GoogleNet	83.54	78.67	75.90	85.89	81.00
VGGNet	82.37	87.98	90.28	79.17	84.95
MDTL-ICDDC	91.77	89.74	93.31	96.66	92.87

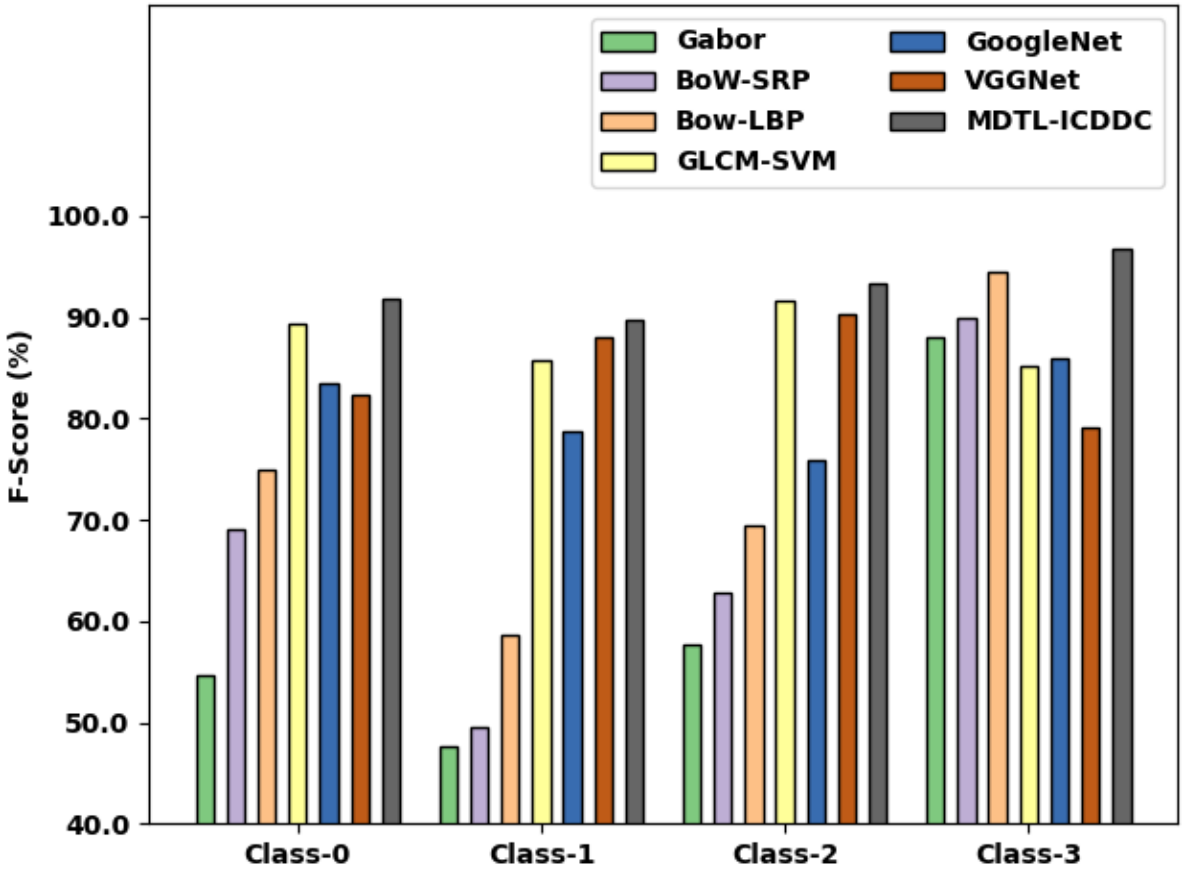

Figure 14. F-score analysis of MDTL-ICDDC technique under various classes.

Moreover, the GoogleNet and VGGNet approaches have resulted in reasonable average F_{score} values of 81% and 84.95% respectively. Finally, the MDTL-ICDDC model has gained superior average F_{score} of 92.87%. These results and discussion pointed out that the MDTL-ICDDC model has proficiently detected and classified the crowd density in enabled video surveillance systems.

5. Conclusions

In this study, a new MDTL-ICDDC method was established for effectual identification and classification of crowd density on video surveillance systems. The MDTL-ICDDC model initially presented an SSA with NASNetLarge model as a feature extraction model in which the hyperparameter tuning process is performed by the SSA. This was followed by the WELM model, which is employed for crowd density and classification processes. At last,

the KSA is applied for the effectual parameter optimization process and thereby improves the classification results. The experimental validation of the MDTL-ICDDC system was carried out with a benchmark dataset and the outcomes are examined under several aspects. The experimental values indicated that the MDTL-ICDDC approach has accomplished enhanced performance over other models. The proposed model can be extended to crowd density estimation in public places such as railway stations, airports, sports stadiums, shopping malls, etc. In the future, the crowd density classification performance can be further boosted by the design of a hybrid metaheuristics algorithm.

Author Contributions: Conceptualization, F.A.; Data curation, F.A. and S.S.A.; Formal analysis, S.S.A.; Investigation, F.N.A.-W. and N.N.; Methodology, N.N. and R.A.; Project administration, R.A.; Software, R.M.; Supervision, R.M.; Validation, A.S.M. and M.A.D.; Visualization, A.S.M. and M.A.D.; Writing—original draft, F.A.; Writing—review & editing, F.N.A.-W. All authors have read and agreed to the published version of the manuscript.

Funding: The authors extend their appreciation to the Deanship of Scientific Research at King Khalid University for funding this work through Large Groups Project under grant number (42/43). Princess Nourah Bint Abdulrahman University Researchers Supporting Project number (PNURSP2022R77), Princess Nourah Bint Abdulrahman University, Riyadh, Saudi Arabia. The authors would like to thank the Deanship of Scientific Research at Umm Al-Qura University for supporting this work by Grant Code: (22UQU4210118DSR22). The authors would like to thank the Deanship of Scientific Research at Majmaah University for supporting this work under Project No. R-2022-xxx.

Institutional Review Board Statement: This article does not contain any studies with human participants performed by any of the authors.

Informed Consent Statement: Not Applicable.

Data Availability Statement: Data sharing was not applicable to this article as no datasets were generated during the current study.

Conflicts of Interest: The authors declare that they have no conflict of interest. The manuscript contains contributions from all authors. All authors have approved the final version of the manuscript.

References

1. Liu, W.; Lis, K.; Salzmann, M.; Fua, P. Geometric and physical constraints for drone-based head plane crowd density estimation. In Proceedings of the 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Macau, China, 3–8 November 2019; IEEE: Piscataway, NJ, USA, 2019; pp. 244–249.
2. Gao, G.; Gao, J.; Liu, Q.; Wang, Q.; Wang, Y. Cnn-based density estimation and crowd counting: A survey. *arXiv* **2020**, arXiv:2003.12783.
3. Fradi, H.; Dugelay, J.L. Towards crowd density-aware video surveillance applications. *Inf. Fusion* **2015**, *24*, 3–15. [CrossRef]
4. Pai, A.K.; Karunakar, A.K.; Raghavendra, U. A novel crowd density estimation technique using local binary pattern and Gabor features. In Proceedings of the 2017 14th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS), Lecce, Italy, 29 August–1 September 2017; IEEE: Piscataway, NJ, USA, 2017; pp. 1–6.
5. Zhou, B.; Song, B.; Hassan, M.M.; Alamri, A. Multilinear rank support tensor machine for crowd density estimation. *Eng. Appl. Artif. Intell.* **2018**, *72*, 382–392. [CrossRef]
6. Sindagi, V.A.; Patel, V.M. A survey of recent advances in cnn-based single image crowd counting and density estimation. *Pattern Recognit. Lett.* **2018**, *107*, 3–16. [CrossRef]
7. Weng, W.T.; Lin, D.T. Crowd density estimation based on a modified multicolumn convolutional neural network. In Proceedings of the 2018 International Joint Conference on Neural Networks (IJCNN), Rio de Janeiro, Brazil, 8–13 July 2018; IEEE: Piscataway, NJ, USA, 2018; pp. 1–7.
8. Anwer, M.H.; Hadeel, A.; Fahd, N.A.; Mohamed, K.N.; Abdelwahed, M.; Anil, K.; Ishfaq, Y.; Abu Sarwar, Z. Fuzzy cognitive maps with bird swarm intelligence optimization-based remote sensing image classification. *Comput. Intell. Neurosci.* **2022**, *2022*, 4063354.
9. Sreenu, G.; Durai, M.S. Intelligent video surveillance: A review through deep learning techniques for crowd analysis. *J. Big Data* **2019**, *6*, 48. [CrossRef]
10. Abunadi, I.; Althobaiti, M.M.; Al-Wesabi, F.N.; Hilal, A.M.; Medani, M.; Hamza, M.A.; Rizwanullah, M.; Zamani, A.S. Federated learning with blockchain assisted image classification for clustered UAV networks. *Comput. Mater. Contin.* **2022**, *72*, 1195–1212. [CrossRef]

11. Sirohi, P.; Al-Wesabi, F.N.; Alshahrani, H.M.; Maheshwari, P.; Agarwal, A.; Dewangan, B.K.; Hilal, A.M.; Choudhury, T. Energy-efficient cloud service selection and recommendation based on QoS for sustainable smart cities. *Appl. Sci.* **2021**, *11*, 9394. [CrossRef]
12. PK, S.; Rabichith, S.N.S.; Borra, S. Crowd density estimation using image processing: A survey. *Int. J. Appl. Eng. Res.* **2018**, *13*, 6855–6864.
13. Lamba, S.; Nain, N. A texture based mani-fold approach for crowd density estimation using Gaussian Markov Random Field. *Multimed. Tools Appl.* **2019**, *78*, 5645–5664. [CrossRef]
14. Ding, X.; He, F.; Lin, Z.; Wang, Y.; Guo, H.; Huang, Y. Crowd density estimation using fusion of multi-layer features. *IEEE Trans. Intell. Transp. Syst.* **2020**, *22*, 4776–4787. [CrossRef]
15. Zhu, L.; Li, C.; Yang, Z.; Yuan, K.; Wang, S. Crowd density estimation based on classification activation map and patch density level. *Neural Comput. Appl.* **2020**, *32*, 5105–5116. [CrossRef]
16. Tang, X.; Xiao, B.; Li, K. Indoor crowd density estimation through mobile smartphone wi-fi probes. *IEEE Trans. Syst. Man Cybern. Syst.* **2018**, *50*, 2638–2649. [CrossRef]
17. Fan, Z.; Zhang, H.; Zhang, Z.; Lu, G.; Zhang, Y.; Wang, Y. A survey of crowd counting and density estimation based on convolutional neural network. *Neurocomputing* **2022**, *472*, 224–251. [CrossRef]
18. Purwar, R.K. Crowd Density Estimation Using Hough Circle Transform for Video Surveillance. In Proceedings of the 2019 6th International Conference on Signal Processing and Integrated Networks (SPIN), Noida, India, 7–8 March 2019; IEEE: Piscataway, NJ, USA, 2019; pp. 442–447.
19. Bouhlel, F.; Mliki, H.; Hammami, M. Abnormal crowd density estimation in aerial images based on the deep and handcrafted features fusion. *Expert Syst. Appl.* **2021**, *173*, 114656. [CrossRef]
20. Li, M.; Chen, T.; Li, Z.; Liu, H. An Efficient Crowd Density Estimation Algorithm Through Network Compression. In *Traffic and Granular Flow 2019*; Springer: Cham, Switzerland, 2020; pp. 165–173.
21. Xiang, J.; Liu, N. Crowd Density Estimation Method Using Deep Learning for Passenger Flow Detection System in Exhibition Center. *Sci. Program.* **2022**, *2022*, 1990951. [CrossRef]
22. Bhuiyan, M.R.; Abdullah, J.; Hashim, N.; Al Farid, F.; Haque, M.A.; Uddin, J.; Isa, W.N.M.; Husen, M.N.; Abdullah, N. A deep crowd density classification model for Hajj pilgrimage using fully convolutional neural network. *PeerJ Comput. Sci.* **2022**, *8*, e895. [CrossRef]
23. Li, Y.C.; Jia, R.S.; Hu, Y.X.; Han, D.N.; Sun, H.M. Crowd density estimation based on multi scale features fusion network with reverse attention mechanism. *Appl. Intell.* **2022**, 1–17. [CrossRef]
24. Wang, S.; Pu, Z.; Li, Q.; Wang, Y. Estimating Crowd Density with Edge Intelligence Based on Lightweight Convolutional Neural Networks. *Expert Syst. Appl.* **2022**, *206*, 117823. [CrossRef]
25. Zaman, K.; Sun, Z.; Shah, S.M.; Shoaib, M.; Pei, L.; Hussain, A. Driver Emotions Recognition Based on Improved Faster R-CNN and Neural Architectural Search Network. *Symmetry* **2022**, *14*, 687. [CrossRef]
26. Mirjalili, S.; Gandomi, A.H.; Mirjalili, S.Z.; Saremi, S.; Faris, H.; Mirjalili, S.M. Salp swarm algorithm: A bioinspired optimizer for engineering design problems. *Adv. Eng. Softw.* **2017**, *114*, 163–191. [CrossRef]
27. Xia, J.; Zhang, H.; Li, R.; Wang, Z.; Cai, Z.; Gu, Z.; Chen, H.; Pan, Z. Adaptive Barebones Salp Swarm Algorithm with Quasi-oppositional Learning for Medical Diagnosis Systems: A Comprehensive Analysis. *J. Bionic Eng.* **2022**, *19*, 240–256. [CrossRef]
28. Utomo, O.K.; Surantha, N.; Isa, S.M.; Soewito, B. Automatic sleep stage classification using weighted ELM and PSO on imbalanced data from single lead ECG. *Procedia Comput. Sci.* **2019**, *157*, 321–328. [CrossRef]
29. Deng, Z.G.; Yang, J.H.; Dong, C.L.; Xiang, M.Q.; Qin, Y.; Sun, Y.S. Research on economic dispatch of integrated energy system based on improved krill swarm algorithm. *Energy Rep.* **2022**, *8*, 77–86. [CrossRef]
30. Meynberg, O.; Cui, S.; Reinartz, P. Detection of high-density crowds in aerial images using texture classification. *Remote Sens.* **2016**, *8*, 470. [CrossRef]
31. Pu, S.; Song, T.; Zhang, Y.; Xie, D. Estimation of crowd density in surveillance scenes based on deep convolutional neural network. *Procedia Comput. Sci.* **2017**, *111*, 154–159. [CrossRef]