Next-Gen Urban Management: Automated Crowd Density Recognition using Rough Neutrosophic Sets for Smart Cities

Eaman Alharbi[1,2,*]

[1]Department of Computer Science, Faculty of Computing and Information Technology, King Abdulaziz University, Jeddah 21589, Saudi Arabia
[2]Center of Research Excellence in Artificial Intelligence and Data Science, King Abdulaziz University, Jeddah 21589, Saudi Arabia
Emails: ealraddadi@kau.edu.sa

Abstract

Neutrosophic set (NS) and Neutrosophic logic (NL) play a major part in approximation theory. They are generalizations of intuitionistic fuzzy sets and logic correspondingly. Rough NS (RNS) combines the concepts of RS and NL to deal with vagueness, uncertainty, and imprecision in information. By integrating truth, indeterminacy, and false degrees, RNS provides a more solid basis for analyzing and classifying complicated data. Particularly, this makes it powerful in applications where incompleteness and ambiguity of data are ubiquitous. Smart cities are a current trend to contain information and communication technologies (ICTs) in the progression of great urban cities. It would be beneficial in defining the city's movement by monitoring the regular flow of traffic jams and visitors. One important characteristic of smart cities is Crowd management, which assists in safety and enjoyable experiences for the residents and visitors. Since the crowd density (CD) classification method encounters tasks including inter-scene, non-uniform density, and intra-scene deviations, occlusion and convolutional neural networks (CNNs) approaches were beneficial. This work focuses on the design of Automated Crowd Density Recognition using the Rough Neutrosophic Set for Smart Cities (ACDR-RNSSC) method in urban management. The presented ACDR-RNSSC method focuses on identifying various types of crowd densities in smart cities. Firstly, the ACDR-RNSSC method utilizes the ResNet50 method for feature extraction. Second, the classification is done using RNS. RNS is utilized for its ability to manage the vagueness and uncertainty in crowd density statistics. Lastly, the parameter is fine-tuned using the Fruit Fly Optimization Algorithm (FOA). This ensures that the model attains high robustness and accuracy in forecasting crowd density. The empirical analysis of the ACDR-RNSSC method is examined under benchmark crowd dataset and the outcomes are tested using various metrics. This study states the improvement of the ACDR-RNSSC method over existing techniques.

Keywords: Neutrosophic Set; Neutrosophic Logic; Crowd Density; Fruit Fly Optimization; ResNet50; Rough Neutrosophic Sets

1. Introduction

The rough sets concept is based on the classifications tool, which is observed as correspondence relations and pieces of knowledge blocks made by it would be a divider of the world [1]. The most interesting theory of generalization of fuzzy sets (FS) and intuitionistic FS (IFS) is the concept of Neutrosophic sets (NS) presented. NSs are defined by 3 functions such as indeterminacy, a membership, and non-membership functions, which are individually connected to each other [2]. While the NSs are a strong tool for dealing with unidentified and variable information, the rough sets concept is a great arithmetic tool for dealing with incompletion. The major goal of this study is to present a unique fusion of intellectual structures named rough NSs [3]. With the fast advancement of smart cities, smart communities, and smart security, weird behavioural scrutiny has become a new point for crowd-event researchers. Nowadays, intellectual monitoring is progressively used in common areas such as hospitals, stations, and campuses. The crowded point is the foremost difficulty among them, which disturbs the investigation

DOI: https://doi.org/10.54216/IJNS.250210
Received: February 08, 2024 Revised: April 29, 2024 Accepted: July 27, 2024

of the weird behavior of the crowds [4]. Mainly it was realized by crowd mass approximation and crowd calculation. They may be grouped into dual types namely crowd crowding analysis-based upon the crowd mass approximation and crowd crowding investigation-based on crowd counting [5].

Crowd analysis has ahead considered focusing between researchers in current times due to some reasons [6]. The huge development in urbanization and the worldwide population managed to increase events likely sporting events, public demonstrations, and political rallies amongst others. Comparable with other computer vision (CV) complications, crowd analysis would have faced numerous difficulties [7]. The insecurity of the problems is for adding a wide range of applications to analyse crowds resulting in expanded focus between researchers in current times. In image processing, CNN algorithms have attained successful results also in the calculation of crowd densities. The latest crowd density estimate methods mainly depend on identification or regression [8]. Recognition techniques may be applied in many cases, which have the least amount of people and no occlusion-like sensors based on nearby frames. Additional approaches that depend on reversion consist of two classes [9]. The first one contains ones that detect handcrafted feature in the images such as the edge and texture features; the next one is the regression function preferred to evaluate accumulated person numbers. The other one depends on density map regression and deep neural network (DNN) and this method is considered the finest for approximating crowd densities [10].

This manuscript focuses on the design of Automated Crowd Density Recognition using the Rough Neutrosophic Set for Smart Cities (ACDR-RNSSC) method in urban management. The presented ACDR-RNSSC method focuses on identifying various types of crowd densities in smart cities. Firstly, the ACDR-RNSSC method utilizes the ResNet50 technique for feature extraction. Second, the classification is done using RNS. RNS is utilized for its ability to manage the vagueness and uncertainty in crowd density statistics. Lastly, the parameter is fine-tuned using the Fruit Fly Optimization Algorithm (FOA). This ensures that the model attains high robustness and accuracy in forecasting crowd density. The empirical analysis of the ACDR-RNSSC method is examined under benchmark crowd dataset and the outcomes are tested using various metrics.

2. Literature Review

In [11], the initial crowd congestion-predicting infrastructure is recommended. This framework contains two major parts: 1) an Ensemble forecasting model, and 2) a decision-making method. The proposed method also assembled the ensemble training models in 3 stages. Considering the first stage, this methodology selects and evaluates various training-based representations, such as Sequence-to-Sequence (Seq2Seq) learning, ARIMA, and DeepSTN. In the 2nd stage, one of the finest 3 techniques executed best in the 1st stage, is chosen to construct the stacked ensemble techniques. Ragab and Sabir [12] developed an Internet of Things assisted deep learning enabled anomaly detection technique for smart city infrastructures, named (IoTAD-SCI) technique. The aim of the proposed IoTAD-SCI technique is to mainly identify the existence of anomalies in smart city environment. The arithmetic optimization algorithm is executed to tune the hyperparameters of the DCN model and ID3 classifier is utilized to classify the identified objects in different classes.

Abdullah et al. [13] presented a unique infrastructure by offering semantic separation techniques. Consider this task, Super-pixels are made by utilizing an enhanced watershed transformation and after limited random fields are applied for obtaining multi-object segmental frames. The next one, the Social Force method is presented for extraction. Later employing the calculated social force, multi-people tracing is done by 3D template union utilizing percentile ranks and non-maximal suppressing. Considered the next, multi-object classification is achieved through deep training Feature Pyramid Networks. Guo et al. [14] proposed a Metro Crowd density estimation Network (named as MCNet) technique. Initially, an Integrating Multi-scale Attention (IMA) component is presented for feature extraction. The invention of the IMA segment is to combine the dilation complication, attention device, and multi-scale feature extractions. In the second level, a unique weightless crowd-textured feature extraction network was presented; it could direct the proceeding of video frame and mechanically remove the texture features for crowd mass approximation.

Wong and Law [15] introduced an infrastructure that associates dimensional and sequential data to mechanically map the routes of single tenants. These frameworks combine rare CCTV video data for creating visual assistances for crowd surveillance, along with an arrangement of crowd mobility graphs (CM Graphs) for storing Chrono spatial features. This outline uses DL-based CV techniques, geometric conversions, and a Kalman filter-based tracing algorithm to automatically retrieve crowd-crowding information. AL-Ghamdi et al. [16] introduced a new deep learning-based Face Mask Detection in Religious Mass Gathering (DLFMD-RMG) technique during the COVID-19 pandemic. The DLFMD-RMG technique focuses mainly on improving the accuracy of detecting facemasks in a crowd religious mass gathering. For face detection, the DLFMD-RMG technique uses YOLOv5 with a ResNet-50 detector.

DOI: https://doi.org/10.54216/IJNS.250210
Received: February 08, 2024 Revised: April 29, 2024 Accepted: July 27, 2024

Garg et al. [17] recommended a new methodology levering DL designs for the separation and classifications of human crowd behavior. This method started to collect inputs from monitoring videos to pick up crowd activities, which are then handled for removing noise and extracting the crowd scenes. The segmental videos are then categorized by applying transmission differential Conjugate Gradient Neural Network (CGNN), improving the accuracy of crowd behavior description. Prezioso et al. [18] presented a unique infrastructure that associates the famous YOLO object recognition algorithms with progressive crowd scrutiny methods, aimed at improving the sympathetic and organization of city crowd movements. The suggested frameworks leverage YOLO's actual time object recognition abilities for detecting several objects inside video frames, with specific attention to detecting persons.

3. The Proposed Model

In this work, we design a new ACDR-RNSSC method in urban management. The aim is to recognize several types of crowd densities in the smart cities. The processes involved in the ACDR-RNSSC model comprise RNS-based classification, ResNet50-based feature extraction, and FOA-based hyperparameter fine-tuning. Figure 1 depicts the workflow of the ACDR-RNSSC technique.

A. Feature Extraction using ResNet50 Model

Firstly, the ACDR-RNSSC method utilizes the ResNet50 technique for feature extraction. The ResNet50 method, one of the TL methods discussed earlier, is employed as a foundation rather than constructing an NN [19]. To deploy the region segmentation to employ create from a pre-trained method instead of the model with no picture data. The dimensional of input images at this layer enhances the count of training and testing data that can be kept in selection memory. Accordingly, 224∗224∗1 has been chosen as the input layer to every parameter. The incoming image must have the outcome of carrying the negative rates to 0.

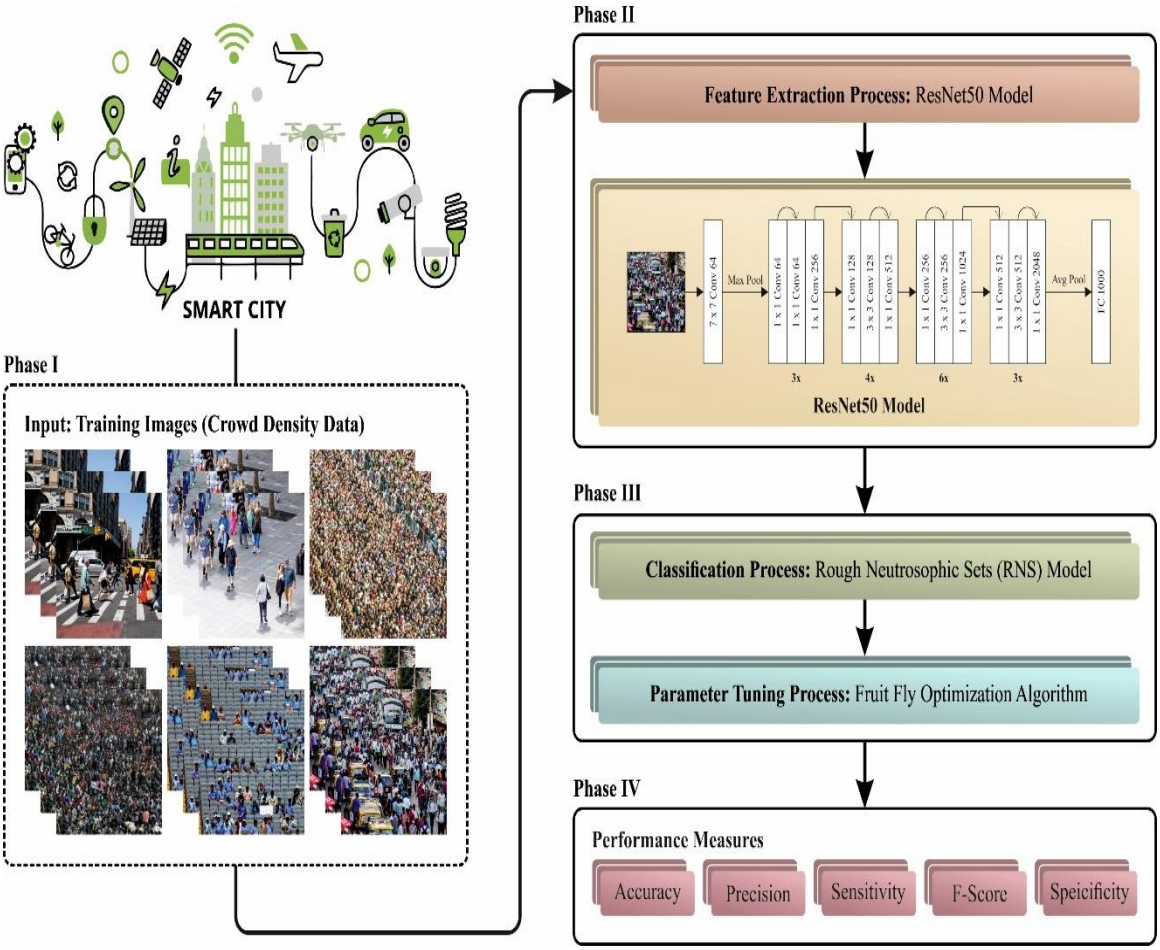

Figure 1. Workflow of ACDR-RNSSC algorithm

DOI: https://doi.org/10.54216/IJNS.250210
Received: February 08, 2024 Revised: April 29, 2024 Accepted: July 27, 2024

The activation function Relu has been executed to forecast. While the Relu layer needs minimal calculating work than the other functions. In Eq. (1), Relu's activation function is as follows:

$$F(x) = \psi\{0, x < 0x, x > 0 \tag{1}$$

The translation layers are fundamental components of the CNN approach. An input image can be filtering maintenance for conversion layers. They selected to filter every data and map image information. All get an N×N size. Eq. (2) offers the curve for this layer that is made up of many linear filters.

$$(h)_{ij} = (W_k + x)_{ij} + b_{ij} \tag{2}$$

At this point i, j refers to the pixel of the index, and W, b denotes the weighted parameters, x signifies the contribution parameter of a feature. The network executes optimum later being normalized. Other layers' data sizes are distinct. It is valuable to normalize the data sizes of other levels. Eqs. (3) and (4) normalized function as:

$$X^k = \frac{x^k - E(x^k)}{\sqrt{var(x^k + \varepsilon}} \tag{3}$$

Whereas, X^k denotes the input parameter, and $E(x^k)$ implies the size $\sqrt{var(x^k + \varepsilon}$ is the standard deviation. γ and βI is the learning variable

$$y^k = \gamma^k x^k + \beta^k \tag{4}$$

Training networks with DL depend on massive data counts that can generate memory proceedings but are in use. It is recommended not to proceed on the internet casually, as it may be necessary to deactivate certain nodes to stop memory issues. By turning off a certain node, memory events can be blocked, and network efficiency is enhanced. The fully connected (FC) layer depends on all the domains within the layer above it and changes the preceding layer's data into a 1D-array design. The count of fully coupled layers varies among systems, with 2 common approaches being deployed max and average pooling. The filters can be elected at the next level at the pool layer. The outcome image dimensional can measured utilizing Eqs. (5)-(6) as:

$$S = W2 * h2 \tag{5}$$

$$W2 = \frac{(W1 - f)}{A + 1} \;\&h2 = \frac{h1 - f}{A + 1} \tag{6}$$

Whereas $w1$ refers to the dimensional of width for image, $h1$ denotes the dimensional of height for image, f represents the dimensional filtering, A implies the number of steps, S signifies the size of input data. Figure 2 represents the structure of ResNet50.

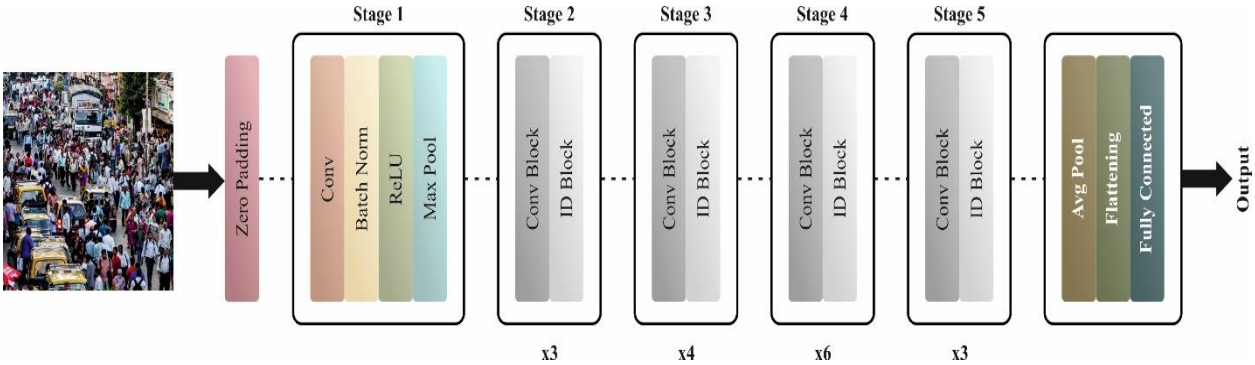

Figure 2. ResNet50 Architecture

At this point, x refers to the count of classes, W denotes the weight, b implies the vector, and the Cross-entropy is provided by Eq. (7). Assume that C_e -cross entropy.

$$c_e = -\sum_x p^1(x) log P(X) \tag{7}$$

Whereas, p' represents the predictable output, p implies the real output for classification.

DOI: https://doi.org/10.54216/IJNS.250210
Received: February 08, 2024 Revised: April 29, 2024 Accepted: July 27, 2024

B. Classification utilizing RNS Technique

The proposed model presented the concept of RNS by merging the neutrosophic sets and rough sets [20]. Intersection, union, inclusion, and equality are some operations. RNS are the generalize of rough FS and rough IFS.

Description 3.1 Assume $\check{R}$ as an equivalence relation and $\check{X}$ as a non-null set. Consider $\check{K}$ as an NS in $\check{X}$ with the functions of indeterminacy $\beta_{\check{K}}$, Membership $\alpha_{\check{K}}$, and non-membership $\gamma_{\check{K}}$. $\overline{A}(\check{K})$ and $\underline{A}(\check{K})$ denotes the upper and lower approximations of $\check{K}$ in $(\check{X},\check{R})$, correspondingly. It is definite as shown in Eq. (8-9).

$$\underline{A}(\check{X}) = \{\{\check{y}: \alpha_{A(\check{K})}(\check{y}), \beta_{A(\check{K})}(\check{y}), \gamma_{\underline{A}(\check{K})}(\check{y})\} | \check{s} \in [\check{y}]_{\check{R}}, \check{y} \in \check{X}\} \quad (8)$$

and

$$\overline{A}(\check{K}) = \{\{\check{y}: \alpha_{\overline{A}(\check{K})}(\check{y}), \beta_{\overline{A}(\check{K})}(\check{y}), \gamma_{\overline{A}(\check{K})}(\check{y})\} | \check{s} \in [\check{y}]_{\check{R}}, \check{y} \in \check{X}\} \quad (9)$$

Let $\underline{A}(\check{K})$ and $\overline{A}(\check{K})$ are dual NSs in $\check{X}$, so $A\overline{T}$ mapping $\overline{A}$, $\underline{A}: \overline{A}(\check{X}) \to \overline{T}(\check{X})$ are mentioned as the inferior and greater RS $A\overline{T}$, and the set $\underline{A}(\check{K}), \overline{A}(\check{K})$ is named as RNS in $(\check{X},\overline{R})$.

The $\underline{A}(\check{K})$ and $\overline{A}(\check{K})$ are continuous degree in the corresponding class of $\check{X}$. When $\underline{A}(\check{K}) = \overline{A}(\check{K})$; then $\alpha_{\underline{A}(\check{K})} = \alpha_{\overline{A}(\check{K})}, \beta_{A(\check{K})} = \beta_{\overline{A}(\check{K})}$ and $\gamma_{\underline{A}(\check{K})} = \gamma_{\overline{A}(\check{K})}$. For any $\check{y} \in \check{X}$, we say $\check{K}$ is a definite NS in the approximation $(\check{X},\check{R})$. It is simply showed that unite NSs $1_{\overline{n}}$ and zero $0_{\overline{n}}$ NSs are definite NSs.

Instance 3.1 Assume $\check{X} = \{\check{h}_1, \check{h}_2, \check{h}_3, \check{h}_4, \check{h}_5, \check{h}_6, \check{h}_7, \check{h}_8\}$ as a universe of discourse. $\check{R}$ as corresponding its part of $\check{X}$ is set as $\check{X}/\check{R} = \{\{\check{h}_1, \check{h}_2\}, \{\check{h}_2, \check{h}_3, \check{h}_6\}, \{\check{h}_5\}, \{\check{h}_7, \check{h}_8\}\}$.

Assume

$\check{A}(\check{K}) = \{(\check{h}_1, (0.3,0.4,0.5)), (\check{h}_4, (0.4,0.6,0.5)), (\check{h}_5, (0.6,0.8,0.4)), (\check{h}_7, (0.3,0.5,0.7))\}$ be an NS of $\check{X}$. By Description of 3.1, we acquire:

$$\overline{A}(\check{K}) = \begin{Bmatrix} (\check{h}_1, (0.4,0.4,0.5)), (\check{h}_4, (0.4,0.4,0.5)), (\check{h}_5, (0.6,0.8,0.4)) \\ (\check{h}_7, (0.3,0.5,0.7)), (\check{h}_8, (0.3,0.5,0.7)) \end{Bmatrix}$$

For another NS,

$$\check{A}(\check{L}) = \{(\check{h}_1, (0.1,0.2,0.3)), (\check{h}_4, (0.1,0.2,0.3)), (\check{h}_5, (0.7,0.6,0.5))\}$$

Then, the upper and lower approximations of $\check{B}(\check{L})$ is expressed as:

$$\overline{A}(\check{L}) = \{(\check{h}_1, (0.1,0.2,0.3)), (\check{h}_4, (0.1,0.2,0.3)), (\check{h}_5, (0.7,0.6,0.5))\},$$

Clearly, $\underline{A}(\check{L}) = \overline{A}(\check{L})$ is a definite NS in the space of approximation $(\overline{X},\overline{R})$.

Description 3.2: When $\overline{A}(\check{K}) = (\underline{A}(\check{L}), \overline{A}(\check{L}))$ is an RNS in $(\hat{X},R)$ and the rough complement of $\check{A}(\check{K})$ is RNS, which is expressed as

by $\sim \check{A}(\check{K}) = (\underline{A}(\check{K})^C, \overline{A}(\check{K})^C)$, whereas $\underline{A}(\check{K})^C$, $\overline{A}(\check{K})^C$ denotes the complements of NSs $\underline{A}(\check{K})$ and $\overline{A}(\check{K})$ correspondingly.

$$\underline{A}(\check{K})^C = \{\{\check{y}, \gamma_{\underline{A}(\check{K})}, (\check{y}), 1 - \beta_{\underline{A}(\check{K})}(\check{y}), \alpha_{\underline{A}(\check{K})}(\check{y})\} | \check{y} \in \check{X}\} \quad (10)$$

And

DOI: https://doi.org/10.54216/IJNS.250210
Received: February 08, 2024 Revised: April 29, 2024 Accepted: July 27, 2024

$$\overline{A}(\breve{K})^{C} = \left\{\left\{\breve{y}, \gamma_{\overline{A}(\breve{K})}, (\breve{y}), 1 - \beta_{\overline{A}(\breve{K})}(\breve{y}), \alpha_{\overline{A}(\breve{K})}(\breve{y})\right\} \middle| \breve{y} \in \breve{X}\right\} \quad (11)$$

Description 3.3 When $\breve{A}(\breve{K}_1)$ and $\breve{A}(\breve{K}_2)$ are dual RNS of the NSs $\overline{K}$ and $\breve{K}$ correspondingly in $\breve{X}$, then it is expressed as:

(a) $\breve{A}(\breve{K}_1) = \breve{A}(\breve{K}_2)$ if and only if $\underline{A}(\breve{K}_1) = \underline{A}(\breve{K}_2)$ and $\overline{A}(\breve{K}_1) = \overline{A}(\breve{K}_2)$

(b) $\breve{A}(\breve{K}_1) \subseteq \breve{A}(\breve{K}_2)$ if and only if $\underline{A}(\breve{K}_1) \subseteq \underline{A}(\breve{K}_2)$ and $\overline{A}(\breve{K}_1) \subseteq \overline{A}(\breve{K}_2)$

(c) $\breve{A}(\breve{K}_1) \cup \breve{A}(\breve{K}_2) = \{\underline{A}(\breve{K}_1) \cup \underline{A}(\breve{K}_2), \overline{A}(\breve{K}_1) \cup \overline{A}(\breve{K}_2)\}$

(d) $\breve{A}(\breve{K}_1) \cap \breve{A}(\breve{K}_2) = \{\underline{A}(\breve{K}_1) \cap \underline{A}(\breve{K}_2), \overline{A}(\breve{K}_1) \cap \overline{A}(\breve{K}_2)\}$

(e) $\breve{A}(\breve{K}_1) + \breve{A}(\breve{K}_2) = \{\underline{A}(\breve{K}_1) + \underline{A}(\breve{K}_2), \overline{A}(\breve{K}_1) + \overline{A}(\breve{K}_2)\}$

(f) $\breve{A}(\breve{K}_1) \cdot \breve{A}(\breve{K}_2) = \{\underline{A}(\breve{K}_1) \cdot \underline{A}(\breve{K}_2), \overline{A}(\breve{K}_1) \cdot \overline{A}(\breve{K}_2)\}$

Description 3.4: Assume that $(\breve{X}, \breve{R})$ as a Pawlak space of approximation, for a value of interval NS,

$\breve{A} = \{< \breve{y}, [\alpha_{\breve{A}}^{\breve{L}}(\breve{y}), \alpha_{\breve{A}}^{\breve{L}}(\breve{y})], [\beta_{\breve{A}}^{\breve{L}}(\breve{y}), \beta_{\breve{A}}^{\breve{L}}(\breve{y})], [\gamma_{\breve{A}}^{\breve{L}}(\breve{y}), \gamma_{\breve{A}}^{\breve{L}}(\breve{y})] > |\breve{y} \in \breve{X}\}$ as an interval NS. The upper and lower approximations $\underline{A}$ of $\breve{A}$ in the Pawlak space of approximation $(\breve{X}, \breve{R})$ are definite as:

$$\underline{A} = \left\{ \begin{matrix} \breve{y}, \left[\Lambda_{\breve{t}\in[\breve{X}]_{\breve{R}}}\left\{\alpha_{\breve{A}}^{\breve{L}}(\breve{y})\right\}, \Lambda_{\breve{t}\in[\breve{X}]_{\breve{R}}}\left\{\alpha_{\breve{A}}^{\breve{U}}(\breve{y})\right\}\right], \left[\nu_{\breve{t}\in[\breve{X}]_{\breve{R}}}\left\{\beta_{\breve{A}}^{\breve{L}}(\breve{y})\right\}, \left[\nu_{\breve{t}\in[\breve{X}]_{\breve{R}}}\left\{\beta_{\breve{A}}^{\breve{U}}(\breve{y})\right\}\right] \right\rangle, \left| \overline{y} \in \overline{X} \right\} \quad (12) \\ \left[\nu_{\breve{t}\in[\breve{X}]_{\breve{R}}}\left\{\gamma_{\breve{A}}^{\breve{L}}(\breve{y})\right\}\right] \vee_{\breve{t}\in[\breve{X}]_{\breve{R}}}\{\gamma_{\breve{A}}^{\breve{U}}(\breve{y})\}] \end{matrix} \right.$$

$$\overline{M}_{\hat{R}} = \{< \breve{y}, [\nu_{\breve{t}\in[\breve{X}]_{\breve{R}}}\{\alpha_{\breve{A}}^{\breve{L}}(\breve{y})\}, \nu_{\breve{t}\in[\breve{X}]_{\breve{R}}}\{\alpha_{\breve{A}}^{\breve{U}}(\breve{y})\}],$$

$$\left[\bigwedge_{\breve{t}\in[\breve{X}]_{\breve{R}}}\left\{\beta_{\breve{A}}^{\breve{L}}(\breve{y})\right\}, \bigwedge_{\breve{t}\in[\breve{X}]_{\breve{R}}}\left\{\beta_{\breve{A}}^{\breve{U}}(\breve{y})\right\}\right],$$

$$\left[\bigwedge_{\breve{t}\in[\breve{X}]_{\breve{R}}}\left\{\gamma_{\breve{A}}^{\breve{L}}(\breve{y})\right\} \bigwedge_{\breve{t}\in[\breve{X}]_{\breve{R}}}\left\{\gamma_{\breve{A}}^{\breve{U}}(\breve{y})\right\}\right] | \overline{y} \in \overline{X}\} \quad (13)$$

C. Hyperparameter Tuning Utilizing FOA Model

Lastly, the parameter selection is fine-tuned using FOA. A metaheuristic model presented the FOA as an outcome of fruit fly foraging performance [21]. Fruit flies organize food when chasing by utilizing their clear insight of smell. While finding the optimizer problem, FOA uses the fruit fly's flight pattern to discover an optimal explanation. In FOA, the fruit fly swarm (possible solution) is first formed randomly in the solution space, and every fruit flier upgrades its location by tracking its method of flying. During the iterative process, the population of fruit flies gradually upsurges its efficacy (rank of solution). One of the smart development calculations is FOA. It is extremely simple to set up, easy to perform, and rapid to expand. However, it also contains some disadvantages. To define the creation scopes of a point throughout the variable implementation, FOA utilizes a random procedure. Dazzle type of equipment is employed throughout natural creation fly individual location upgrade. Presently, numerous estimation principles are accessible for the performance of classifiers. Estimations for classifier reform constantly utilize classifier accuracy and rate of error as vital methods.

FOA is used to enhance the hyperparameter by generating a population of fruit flies, which signifies probable solution in the space of solution. Over iterations, every fruit fly upgrades its location dependent on fitness, progressively enhancing the complete efficacy of the population. This procedure ensures until the global optimal, signifying the finest set of hyperparameter is attained, improving the performance of models.

DOI: https://doi.org/10.54216/IJNS.250210
Received: February 08, 2024 Revised: April 29, 2024 Accepted: July 27, 2024

The FOA enhances by imitating fruit fly behavior of foraging. This bio-inspired technique delivers benefits in compound parameter space, presenting an effectual and quicker convergence towards global goals. The random group permits effectual search, creating it useful for enhancing variables.

Stage1: Initialize the input variables for FOA and include the highest iteration count, dimension of population, lower and upper limit variables, and problem size.

Stage2: Depending upon the upper and lower restraints of the factor, the location of the fruit fly group is formed randomly.

Stage3: Reliant on the fruitfly location, the main population for FOA was made.

Stage4: Use location of fruit fly, evaluate the fitness of complete populace of fruit flies.

Stage5: Signify a common optimal, where the finest fruit fly is positioned.

Stage6: Alter every fruit fly's position in the swarming procedure, and evaluate every fly's fitness.

Stage7: If certainly, the fitness of the highest fly in the group is greater than the global optimum, then update the global optimal.

Stage8: Adjust the iteration $v, v - v + 1, v$, if it is larger than the highest iteration count, or else, go back to stage 6.

Stage9: Provide the best parameter pair (F, μ) as the global optimal.

The FF is the extensive feature influencing the efficiency of the FOA technique. The hyperparameter range technique covers the solution-encoded system to evaluate the candidate solution efficiency. In this work, accuracy is the foremost criterion and considered by FOA to design the FF that is stated as.

$$Fitness = \max(P) \tag{14}$$

$$P = \frac{TP}{TP + FP} \tag{15}$$

Here, TP and FP denotes the true and false positive value, correspondingly.

4. Experimental Validation

The simulation analysis of the ACDR-RNSSC approach is implemented utilizing a crowd dataset with 1000 instances as denoted in Table 1. The database has 250 instances under each class.

Table 1: Details on dataset

Labels	Classes	Sample count
Class1	Dense Crowd	250
Class2	Medium Dense Crowd	250
Class3	Sparse Crowd	250
Class4	No Crowd	250
Total No. of Samples		1000

DOI: https://doi.org/10.54216/IJNS.250210
Received: February 08, 2024 Revised: April 29, 2024 Accepted: July 27, 2024

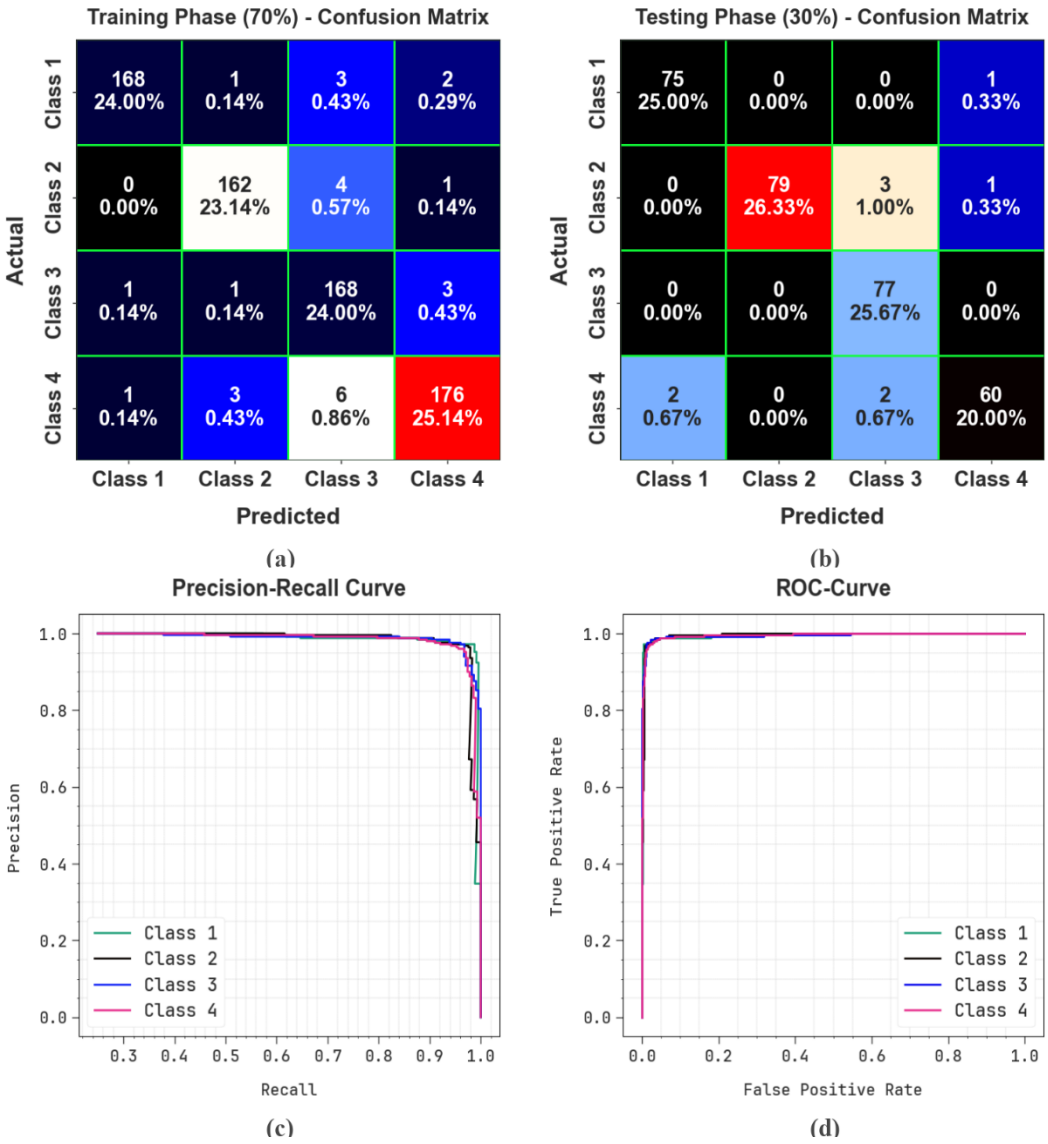

Figure 3. Classifier outcome of (a-b) Confusion matrices and (c-d) PR and ROC curves

Figure 3 validates the classifier results of the ACDR-RNSSC model in the test database. Figures. 3a-3b signifies the confusion matrices presented by the ACDR-RNSSC methodology on 70:30 of TRAP/TESP. The outcome indicated that the ACDR-RNSSC method has recognized and categorized every classes. In addition, Figure 3c authorizes the PR curve of the ACDR-RNSSC model. The results specified that the ACDR-RNSSC model has enlarged the maximum efficiency of PR under every class. At last, Figure 3d demonstrates the study of ROC of the ACDR-RNSSC model. The outcomes signified that the ACDR-RNSSC method has stemmed in expert outcomes with the uppermost value of ROC below different class labels.

In Table 2, a detailed recognition result of the ACDR-RNSSC method is represented for 70:30 of TRAP/TESP. The tabulated values inferred that the ACDR-RNSSC technique proficiently recognizes four class labels.

The average recognition results of the ACDR-RNSSC technique on 70 %TRAP is established in Figure 4. The outcomes specify that the ACDR-RNSSC approach can efficaciously identify the classes. It is also perceived that the ACDR-RNSSC system gets an average $accu_y$, $prec_n$, $sens_y$, $spec_y$, and F_{score} of 98.14%, 96.34%, 96.32%, 98.76%, and 96.31%, respectively.

DOI: https://doi.org/10.54216/IJNS.250210
Received: February 08, 2024 Revised: April 29, 2024 Accepted: July 27, 2024

Table 2: Classifier Outcomes of ACDR-RNSSC method under 70%TRAP and 30%TESP

Class Labels	$Accu_y$	$Prec_n$	$Sens_y$	$Spec_y$	F_{Score}
TRAP (70%)					
Class 1	98.86	98.82	96.55	99.62	97.67
Class 2	98.57	97.01	97.01	99.06	97.01
Class 3	97.43	92.82	97.11	97.53	94.92
Class 4	97.71	96.70	94.62	98.83	95.65
Average	98.14	96.34	96.32	98.76	96.31
TESP (30%)					
Class 1	99.00	97.40	98.68	99.11	98.04
Class 2	98.67	100.00	95.18	100.00	97.53
Class 3	98.33	93.90	100.00	97.76	96.86
Class 4	98.00	96.77	93.75	99.15	95.24
Average	98.50	97.02	96.90	99.00	96.92

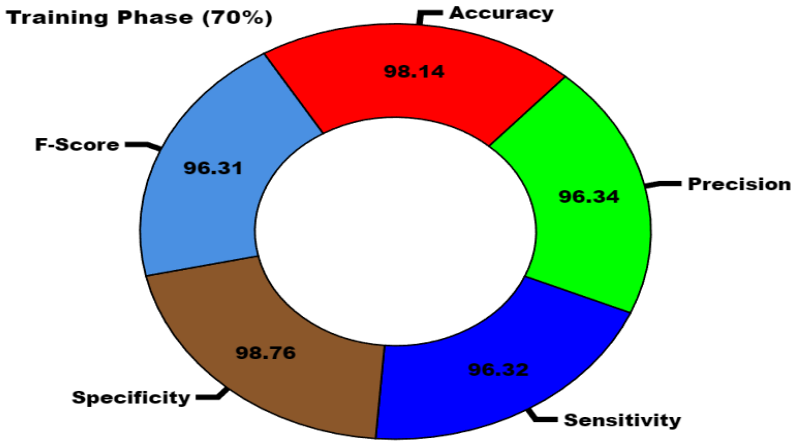

Figure 4. Average outcome of ACDR-RNSSC method under 70% TRAP

The average detection outcome of the ACDR-RNSSC technique on 30%TESP is shown in Figure 5. The outcomes specify that the ACDR-RNSSC model can efficiently recognize the classes. It is also noted that the ACDR-RNSSC method gets an average $accu_y$, $prec_n$, $sens_y$, $spec_y$, and F_{score} of 98.50%, 97.02%, 96.90%, 99.00%, and 96.92%, respectively.

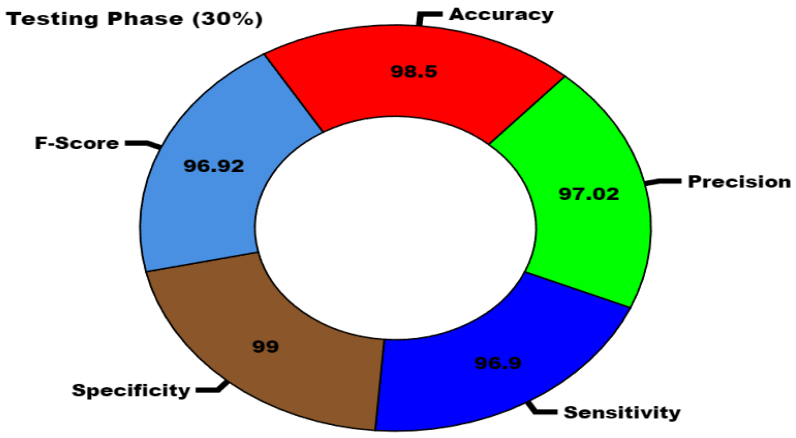

Figure 5. Average outcome of ACDR-RNSSC method under 30% TESP

DOI: https://doi.org/10.54216/IJNS.250210
Received: February 08, 2024 Revised: April 29, 2024 Accepted: July 27, 2024

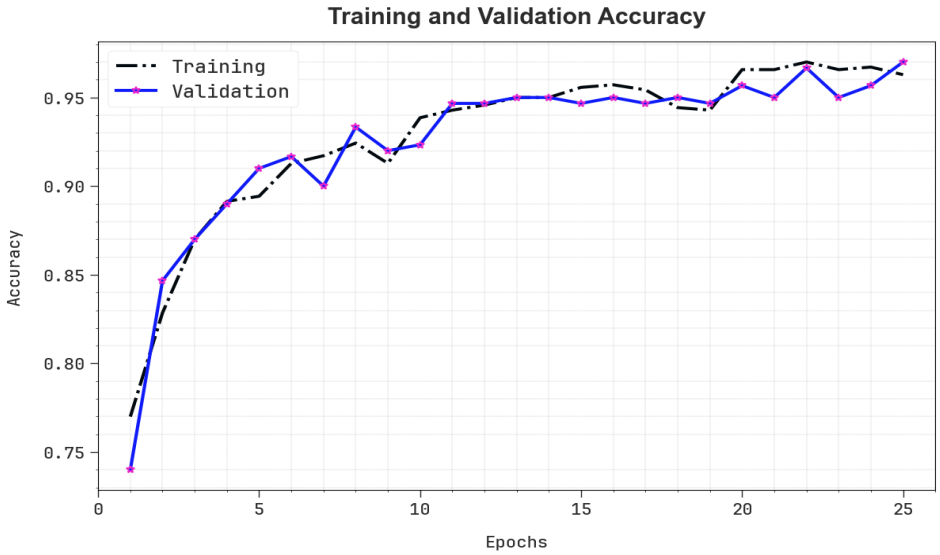

Figure 6. $Accu_y$ Curve of ACDR-RNSSC technique

In Figure 6, the training and validation accuracy results of the ACDR-RNSSC method have been established. The accuracy values are computed over a range of 0-25 epochs. The outcome highlighted that the training and validation accuracy values show a rising tendency, which alerted the ability of the ACDR-RNSSC technique with improved performance over many iterations. Furthermore, the training and validation accuracy remains closer over the epochs, which labels last minimal overfitting and displays the amended performance of the ACDR-RNSSC method, assuring constant prediction on unseen samples.

Figure 7. Loss curve of ACDR-RNSSC technique

In Figure 7, the training and validation loss graph of the ACDR-RNSSC system is exhibited. The loss values are calculated over a range of 0-25 epochs. It is expressed that the training and validation accuracy values establish a declining tendency, which shows the capability of the ACDR-RNSSC technique in balancing a trade-off between data fitting and generalization. The continual reduction in loss values guarantees the amended performance of the ACDR-RNSSC technique and tunes the prediction outcomes over time.

In Table 3 and Figure 8, the experimental outcomes of the ACDR-RNSSC methodology with current techniques are given [22]. The results exhibit that the BoWSRP model has shown worse performance with $accu_y$, $prec_n$, $reca_l$, and F_{score} of 79.95%, 68.25%, 67.75%, and 84.76%, respectively. Simultaneously, the GLCM-SVM technique has obtained slightly increased outcomes with $accu_y$, $prec_n$, $reca_l$, and F_{score} of 80.20%, 75.57%, 74.37%, and 74.74%, respectively. Besides, the BoWLBP, GoogleNet, VGGNet, and MDTL-ICDDC techniques

DOI: https://doi.org/10.54216/IJNS.250210
Received: February 08, 2024 Revised: April 29, 2024 Accepted: July 27, 2024

have attained moderately closer performance. Meanwhile, the AOTL-CDA3S system has resulted in considerable outcomes with $accu_y$, $prec_n$, $reca_l$, and F_{score} of 98.00%, 96.11%, 95.87%, and 95.93%, respectively. But the ACDR-RNSSC technique outperforms the other models with maximum $accu_y$, $prec_n$, $reca_l$, and F_{score} of 98.50%, 97.02%, 96.90%, and 96.92%, respectively.

Table 3: Comparative outcome of ACDR-RNSSC technique with existing methods

Models	$Accu_y$	$Prec_n$	$Reca_l$	F_{score}
ACDR-RNSSC	98.50	97.02	96.90	96.92
AOTL-CDA3S	98.00	96.11	95.87	95.93
VGGNet	84.49	85.72	82.74	84.76
BoWSRP	79.95	68.25	67.75	68.32
BoWLBP	84.26	74.36	74.37	74.74
GLCM-SVM	80.20	75.57	73.63	87.99
GoogleNet	84.51	83.16	84.94	80.92
MDTL-ICDDC	96.94	93.24	93.11	93.29

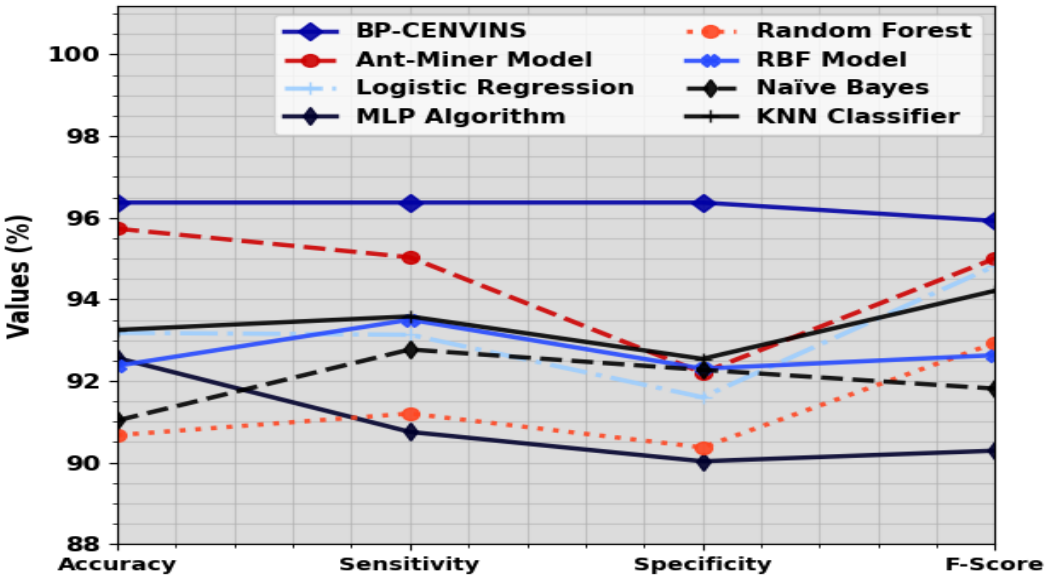

Figure 8. Comparative outcome of ACDR-RNSSC technique with existing methods

5. Conclusion

In this text, we design a new ACDR-RNSSC methodology in urban management. The presented ACDR-RNSSC technique focuses on detecting several types of crowd densities in smart cities. Initially, the ACDR-RNSSC method employs the ResNet50 model for feature extraction. Second, the classification is performed using RNS. RNS is employed for its capacity to manage the vagueness and uncertainty in crowd density statistics. Finally, the parameter is fine-tuned by utilizing the FOA method. This ensures that the model attains high robustness and accuracy in forecasting crowd density. The empirical findings of the ACDR-RNSSC methodology is investigated under a benchmark crowd dataset and the results are tested using various metrics.

Funding: The author gratefully acknowledges technical support provided by the Center of Research Excellence in Artificial Intelligence and Data Science, King Abdulaziz University, Jeddah 21589, Saudi Arabia. Faculty of Computing and Information Technology and Faculty of Tourism at King Abdulaziz University, Jeddah, Saudi Arabia"

Conflicts of Interest: "The authors declare no conflict of interest."

DOI: https://doi.org/10.54216/IJNS.250210
Received: February 08, 2024 Revised: April 29, 2024 Accepted: July 27, 2024

References

[1] Al-Hamido, R.K., Salha, L. and Gharibah, T., 2020. Neutrosophic crisp semi separation axioms in neutrosophic crisp topological spaces. International Journal of Neutrosophic Science, 6(1), pp.32-40.

[2] Abobala, M., 2020. Classical homomorphisms between refined neutrosophic rings and neutrosophic rings. International Journal of Neutrosophic Science, 5, pp.72-75.

[3] Das, S.K. and Edalatpanah, S.A., 2020. A new ranking function of triangular neutrosophic number and its application in integer programming. International Journal of Neutrosophic Science, 4(2), pp.82-92.

[4] Dhar, M., 2020. Neutrosophic soft block matrices and some of its properties. Int J Neutrosophic Sci, 12(1), pp.39-49.

[5] Chinnadurai, V. and Sindhu, M.P., 2020. An introduction to neutro-fine topology with separation axioms and decision-making. International Journal of Neutrosophic Science (IJNS) Volume 12, 2020, p.11.

[6] Minoura, H.; Yonetani, R.; Nishimura, M.; Ushiku, Y. Crowd density forecasting by modeling patch-based dynamics. IEEE Robot. Autom. Lett. 2020, 6, 287–294.

[7] Fitwi, A.; Chen, Y.; Sun, H.; Harrod, R. Estimating interpersonal distance and crowd density with a single-edge camera. Computers 2021, 10, 143.

[8] 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.

[9] 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.

[10] Almagbile, A. Estimation of crowd density from UAVs images based on corner detection procedures and clustering analysis. Geo-Spat. Inf. Sci. 2019, 22, 23–34.

[11] Derhab, A., Mohiuddin, I., Halboob, W. and Almuhtadi, J., 2024. Crowd Congestion Forecasting Framework using Ensemble Learning Model and Decision making Algorithm: Umrah use case. IEEE Access.

[12] Ragab, M. and Sabir, M.F.S., 2022. Arithmetic optimization with deep learning enabled anomaly detection in smart city. CMC-Computers Materials & Continua, 73(1), pp.381-395.

[13] Abdullah, F., Abdelhaq, M., Alsaqour, R., Alatiyyah, M.H., Alnowaiser, K., Alotaibi, S.S. and Park, J., 2023. Context aware crowd tracking and anomaly detection via deep learning and social force model. IEEE Access.

[14] Guo, Q., Zhang, R. and Zhao, D., 2024. MCNet: A crowd denstity estimation network based on integrating multiscale attention module. arXiv preprint arXiv:2403.20173.

[15] Wong, V.W. and Law, K.H., 2023. Fusion of CCTV Video and Spatial Information for Automated Crowd Congestion Monitoring in Public Urban Spaces. Algorithms, 16(3), p.154.

[16] AL-Ghamdi, A.S., Alshammari, S.M. and Ragab, M., 2023. Deep Learning Based Face Mask Detection in Religious Mass Gathering During COVID-19 Pandemic. Computer Systems Science & Engineering, 46(2), pp. 1863-1877.

[17] Garg, S., Sharma, S., Dhariwal, S., Priya, W.D., Singh, M. and Ramesh, S., 2024. Human crowd behaviour analysis based on video segmentation and classification using expectation–maximization with deep learning architectures. Multimedia Tools and Applications, pp.1-23.

[18] Prezioso, E., Giampaolo, F., Izzo, S., Savoia, M. and Piccialli, F., 2023, October. Integrating Object Detection and Advanced Analytics for Smart City Crowd Management. In 2023 IEEE International Conference on Networking, Sensing and Control (ICNSC) (Vol. 1, pp. 1-6). IEEE.

[19] Jayashree, S. and Sumalatha, V., 2024, June. Plant Leaf Disease Detection Using Resnet-50 Based on Deep Learning. In International Conference on Digital Transformation in Business: Navigating the New Frontiers beyond Boundaries (DTBNNF 2024) (pp. 150-166). Atlantis Press.

[20] Smarandache, F., Ali, M. and Khan, M., 2019. Arithmetic operations of neutrosophic sets, interval neutrosophic sets and rough neutrosophic sets. Fuzzy Multi-criteria Decision-Making Using Neutrosophic Sets, pp.25-42.

[21] Sonia, R., Joseph, J., Kalaiyarasi, D., Kalyani, N., Gopala Gupta, A.S., Ramkumar, G., Almoallim, H.S., Alharbi, S.A. and Raghavan, S.S., 2024. Segmenting and classifying skin lesions using a fruit fly optimization algorithm with a machine-learning framework. Automatika, 65(1), pp.217-231.

[22] Al Duhayyim, M., Alabdulkreem, E., Tarmissi, K., Aljebreen, M., El Khier, B.S.I.A., Zamani, A.S., Yaseen, I. and I. Eldesouki, M., 2022. Aquila optimization with transfer learning based crowd density analysis for sustainable smart cities. Applied Sciences, 12(21), p.11187.

DOI: https://doi.org/10.54216/IJNS.250210
Received: February 08, 2024 Revised: April 29, 2024 Accepted: July 27, 2024