

# Multilevel Thresholding for Multi-Spectral Image Using Convolutional Fuzzy Clustering Algorithm and Gradient Multilayer Kernelized Perceptron

Mr. M. Arun Prasad<sup>1</sup>, Dr. N. P. Subiramaniam<sup>2</sup>

Submitted: 21/03/2023

Revised: 25/05/2023

Accepted: 10/06/2023

**Abstract:** When dealing with the issue of low-dimensional images, a multispectral image is made up of many bands with high dimensions. Both in terms of accuracy and calculation time, the current multilevel thresholding approaches are ineffective. Although they require a lot of work, 2D histogram-based approaches are better in terms of accuracy. This study suggests a unique method for segmenting and classifying various images based on multilayer thresholding and deep learning algorithms. Here, a various image is used as the input, and it has undergone noise removal, smoothing, and image resizing processes. Processed image has been segmented utilizing convolutional operation based fuzzy clustering with multilevel thresholding (Con\_Fuz\_Clus\_MT) of the input image. Then this image has been classified using gradient multilayer Kernelized perceptron integrated with Darwinian optimization (GMKP-DO). Experimental findings confirmed by statistical analysis show that the newly created approach can produce accurate predictions. We demonstrate that combining two approaches—threshold-based and DL enhances cloud identification performance without requiring manual correction of automatically generated ground facts. Proposed technique attained classification accuracy of 93%, precision of 88%, recall of 85%, F-measure of 89%, ROC of 75% for flight dataset; Classification accuracy of 94%, precision of 92%, recall of 90%, F-measure of 90%, ROC of 41% for bird image dataset; Classification accuracy of 96%, precision of 82%, recall of 65%, F-measure of 56%, ROC of 45% for car image dataset.

**Keywords:** Multispectral image, multilevel thresholding, segmentation, classification, deep learning.

## 1. Introduction:

As remote sensing (RS) technology has advanced, RS platforms have gotten better at gathering a variety of data. By identifying changes to the ground surface, these data have evolved into important resources for environmental monitoring. Change detection (CD) is a phenomena that involves identifying changes in a given geographic area by comparing a collection of photographs taken at various times [1]. As a result, academics from throughout the world are paying more attention to CD. Limited temporal, spatial, and spectral resolutions in RS data place severe limitations on RS-based CD techniques. However, many of these limitations have been solved by the development of sensors with higher technical capabilities. Researchers have therefore looked at an ever-growing range of approaches, algorithms, and processes for identifying change [2]. There have been several different kinds of satellites put into orbit in RS, including active and passive, optical or microwave sensors, and high- or low-

resolution ones. Many complicated tasks, including object detection, localization, segmentation, and interpretation of natural scenes, have seen considerable improvements in terms of efficiency, speed, and scalability, especially following the implementation of DL methods like CNN [3]. However, there is a need for alternative imaging systems if conditions like this have a good probability of occurring, such as in a foggy environment or one with poor illumination. RGB systems can't offer enough information in these situations [4]. Another such dependable tool is thermal infrared imaging, which records heat signatures and enables infrared data to be independent of the effectiveness of the light source (s). The accurate recognition of existence, location, and type of structural problems are typical objectives of structure health monitoring. Different evaluation models can use these pieces of data to guide judgments about rehabilitation strategies to extend service life of infrastructure components [5]. Corrosion has a negative impact on the functionality and durability of metal infrastructure components. According to reports, corrosion accounts for 42% of all failure mechanisms in engineering structures, making it the most common type of defect [6].

Contribution of this research is as follows:

<sup>1</sup>Research Scholar, Department of Electronics, Nehru Arts and Science College, Coimbatore, Tamil Nadu, India. Email ID: arunprasad1012@gmail.com

<sup>2</sup>Professor, Department of Electronics, Nehru Arts and Science College, Coimbatore, Tamil Nadu, India. Email ID: nalsunil@gmail.com

1. To propose unique method for segmenting and classifying multi-spectral images based on multilayer thresholding and deep learning algorithms.
2. Processed image has been segmented utilizing convolutional operation based fuzzy clustering with multilevel thresholding (Con\_Fuz\_Clus\_MT) of the input image.
3. This image has been classified using gradient multilayer Kernelized perceptron integrated with Darwinian optimization (GMKP-DO).

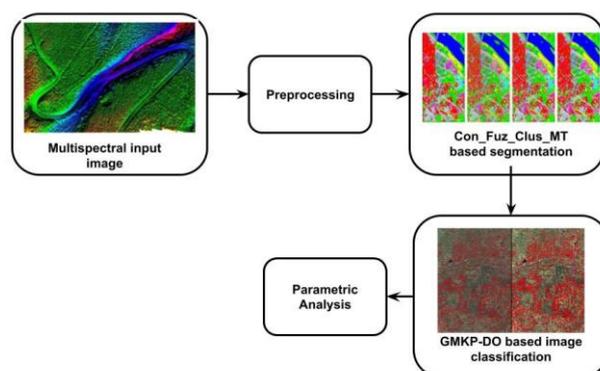
## 2. Related Works:

Aspects (colour, texture, and shape) that are retrieved from digital photographs are used by [7]. In order to distinguish corroded areas from metal surfaces, data dimensionality reduction and linear classifiers are also used. The complexity of the relevant problem, however, makes it possible to improve accuracy of aforementioned detection method with nonlinear as well as complicated data classification techniques. For the purpose of stochastic modelling of pitting corrosion that appears on metallic structures, work [8] created Markov chain models. The usefulness of grey level co-occurrence matrix as well as colour descriptors in classifying corroded and non corroded surfaces was established by [9]. Visual inspection is used in [10] to identify pitting corrosion using high dynamic range images. Work [11] usage Methods for colour image processing and Fourier transform-based fault detection on steel bridge surfaces. A computer vision-based technique for classifying corrosion flaws on coated materials utilizing watershed segmentation technique was proposed by [12]. An automated method for rust defect detection in steel bridge coatings was developed by [13] and is based on Image texture analysis as well as DFT. The effectiveness of corrosion detection was examined by [14] in relation to colour space, colour channels, and subimage block size. For the purpose of identifying problematic areas,

this method also used colour wavelet-based texture analysis methods [15]. The majority of current tree segmentation methods are based on user-defined methods that explain how trees appear in a hierarchical order of rules. These methods can be roughly divided into edge detection methods, valley following methods, local maxima-minima methods, contour detection methods, region growth methods, template matching methods, watershed routines, and watershed routines. A recent comparative study [16] found that utilizing 3D segmentation from LIDAR data rather than relying on surface RGB images significantly improves crown segmentation. When they are combined, more advancement is anticipated. DL as well as learning-based methodologies [17] also shows commendable outcomes. Convolutional neural networks (CNNs) are used in a recent semi-supervised method [18] that mixes LIDAR and RGB data to produce results that are comparable to those of traditional unsupervised algorithms. Data from multi-spectral imaging was also employed with CNNs [19]. A deep network was used in [20] to distinguish between trees, bare soil, and weeds. Despite the fact that they deliver reliable results, they require a substantial amount of training data.

## 3. Proposed Methodology:

This research proposes novel method in various image analysis based on multilevel thresholding based on segmentation with classification using DL techniques. Here input image is taken as multispectral image and it has been processed for noise removal, smoothing, image resize. Processed image is segmented using convolutional operation based fuzzy clustering with multilevel thresholding (Con\_Fuz\_Clus\_MT) of the input image. Then this image has been classified using gradient multilayer Kernelized perceptron integrated with Darwinian optimization (GMKP-DO). Proposed architecture is represented in figure-1.



**Fig 1** Proposed architecture of various image analyses in multilevel thresholding

Let's assume that an image has L intensity levels, or three colour components for RGB images, with each level falling between 0 and L-1. After that, one may define by eq. (1)

$$p_i^C = \frac{h_i^C}{N} \quad \sum_{i=0}^{L-1} p_i^C = 1 \quad (1)$$

where N denotes overall number of pixels in image; In other words, probability distribution is thought of as the normalised representation of an image histogram for every component C represented by hci. It is simple to determine the combined mean, or total mean, of every component of image as eq. (2):

$$\mu_T^C = \sum_{i=0}^{L-1} i p_i^C = 1 \quad (2)$$

Operation is carried out as follows: n-level thresholding presents n-1 threshold levels at  $t_j^C, j = 1, \dots, n-1$  by eq. (3).

$$F^C(x, y) = \begin{cases} 0, & f^C(x, y) \leq t_1^C \\ \frac{1}{2}(t_1^C + t_2^C), & t_1^C < f^C(x, y) \leq t_2^C \\ \vdots & \\ \frac{1}{2}(t_{n-2}^C + t_{n-1}^C), & t_{n-2}^C < f^C(x, y) \leq t_{n-1}^C \\ L-1, & f^C(x, y) > t_{n-1}^C \end{cases} \quad (3)$$

In this case, n classes ( $D_c 1, \dots, D_c n$ ) of pixels from a given image is created, each of which may represent one or more objects or even a subset of those items (e.g., topological features). The following table lists the probabilities of occurrence for classes  $D_c 1, \dots, D_c n$  by eq. (4).

$$w_j^C = \begin{cases} \sum_{i=0}^{t_j^C} p_i^C, & j = 1 \\ \sum_{i=t_{j-1}^C+1}^{t_j^C} p_i^C, & 1 < j < n \\ \sum_{i=t_{j-1}^C+1}^{L-1} p_i^C, & j = n. \end{cases} \quad (4)$$

Then, it is possible to get the mean for each class  $\mu_j^C$  as eq. (5)

$$\mu_j^C = \begin{cases} \sum_{i=0}^{t_j^C} \frac{p_i^C}{w_j^C}, & j = 1 \\ \sum_{i=t_{j-1}^C+1}^{t_j^C} \frac{p_i^C}{w_j^C}, & 1 < j < n \\ \sum_{i=t_{j-1}^C+1}^{L-1} \frac{p_i^C}{w_j^C}, & j = n. \end{cases} \quad (5)$$

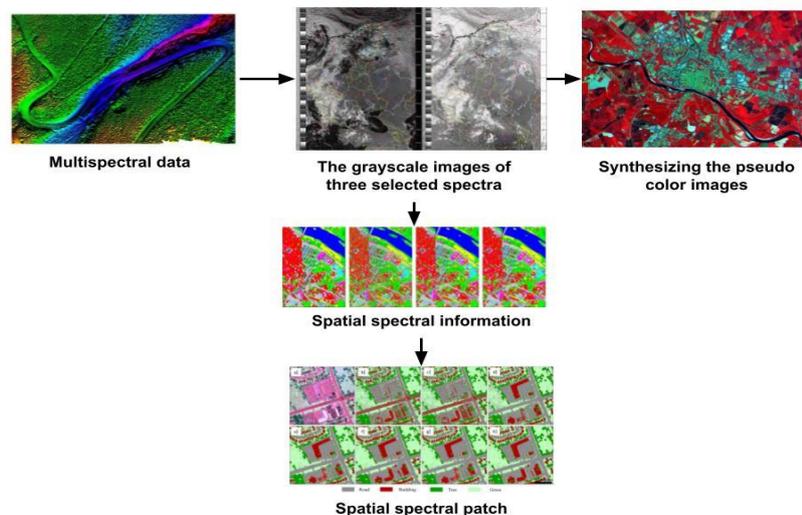
The approach to determining the ideal threshold that optimises between-class variance of every component, which is commonly characterised as eq. (6)

$$\sigma_B^{C^2} = \sum_{j=1}^n w_j^C (\mu_j^C - \mu_T^C)^2 \quad (6)$$

$$\varphi^C = \max_{1 < t_1^C < \dots < t_{n-1}^C} \sigma_B^{C^2}(t_j^C) \quad (7)$$

### Convolutional operation based fuzzy clustering with multilevel thresholding:

A 3x3 kernel is convolved with layer's input in a convolution layer. Fig. 2 displays the proposed network's block diagram. It includes five blocks for decoding and six for encoding. Two convolution layers are used in each of these blocks to extract the image's semantic information. To produce the output, ReLU [10] is then used. Using a copy layer, the analogous decoding block makes advantage of image features obtained from encoding block.



**Fig-2** CNN based multi-spectral image analysis

The proposed network's input images have spatial dimensions of  $192 \times 192 \times 4$ . We must divide every Landsat 8 spectral band into smaller image patches because they are all quite large—on the order of  $8000 \times 8000$  pixels. The result is the cropping of each spectral band picture into  $384 \times 384$  non-overlapping patches. These patches are reduced in size to  $192 \times 192$  prior to training. The 4D input is then supplied to network. Output probability map is extracted using a sigmoid activation function in network's final convolution layer. Utilizing the Adam gradient descent method, following soft Jaccard loss function is built to optimise network by eq. (8):

$$L(h, y) = - \frac{\sum_{i=1}^n h_i y_i + \epsilon}{\sum_{i=1}^n h_i + \sum_{i=1}^n y_i - \sum_{i=1}^n h_i y_i + \epsilon}, \quad (8)$$

Here, the output of sigmoid function in network's last layer yields probability map, which is denoted by letters  $y$  and  $h$  respectively. The ground truth's total number of pixels is  $n$ . We chose a learning rate for the training that was initially set at 104. There are 600 epochs in training method. Network converges to a suitable local minimum after this number of epochs. The derived weights are then put to use in order to make predictions. These patches are then scaled down to  $192 \times 192$  and layered. The resultant cloud probability map is enlarged to  $384 \times 384$  pixels once a cloud feature corresponding to every patch is acquired.

Assume that  $I$  represents the data set of apple photographs, with  $I = [\text{Image1}, \text{Image2}, \text{Image3}, \text{Image4}, \dots, \text{ImageN}]$ , where  $N$  is total number of images in data collection. The region of interest is extracted using Algorithm 3 for each  $\text{Image}_i$ , where  $I = 1 \dots N$ . The dimensions of each  $\text{Image}_i$  will be  $R \times G \times B$ , where  $R$  stands for red channel,  $G$  for green, and  $B$  for blue. The most pixel-dense cluster,  $\text{Clust}_i$ , was used to determine threshold value, " $T$ ," as specified in Algorithm 1 by eq. (9):

$$\text{Sim}_{\text{image}} = \begin{cases} 0 & \text{if Sim}_{\text{image}}(m, n) > T \\ 1 & \text{if Sim}_{\text{image}}(m, n) < T \end{cases} \quad (9)$$

One of the unsupervised clustering techniques is the Fuzzy C-Means (FCM) method. Due to its straightforward functionality, this strategy is one of the most popular. For a set of features with  $f_j \in \mathbb{R}$ ,  $j = 1, 2, \dots, C$ , existing FCM is utilised to discover an image partition with 'p' and fuzzy clusters while minimising the cost function by eq. (10).

$$J(U, M) = \sum_{i=1}^p \sum_{j=1}^c (U_{i,j})^m d_{i,j} \quad (10)$$

Here,  $M = [m_1, \dots, m_p]$  refers to a cluster centre matrix. Additionally, it is noted that the word  $m \in [1, \infty]$  is a fuzzification parameter. Following are the operational steps of proposed fuzzy cluster based transition feature variance based segmentation method:

- a) Set the appropriate initial values for variables like  $m$ ,  $c$ , and a tiny positive integer  $\epsilon$ .
- b) For periods  $t_1$  and  $t_2$ , select the cluster centre at random.
- c) Let consider variable  $t = 0$ .
- d) Determine partition matrix's fuzzy value at time  $t = 0$  between  $t_1$  and  $t_2$ .
- e) Change fuzzy partition matrix  $u$  as well as the value  $t > 0$  to by eq. (11)

$$u^{(t+1)}(t_1, t_2) = \frac{1}{\left( \sum_{l=1}^p \left( \frac{d_{lj}}{d_{li}} \right)^{1/(1-m)} \right)} (t_1, t_2) \quad (12)$$

For  $i = 1, \dots, p$  and  $j = 1, \dots, C$

- f) Update  $t$  value at  $t = 0$  or update condition is satisfied by eq. (13):

$$\sigma_b^2(t) = \sigma^2 - \sigma_\omega^2(t) \quad (13)$$

The following [(1)–(6)] describes an exhaustive search for threshold that reduces intra-class variance, which is described as a weighted sum of variance of 2 classes. :

1. Create a grayscale version of the RGB image.
2. Evaluate histogram and odds for every intensity level.
3. Set up initial  $\omega_i(0)$  and  $\mu_i(0)$ .
4. Step past every threshold at its highest level of intensity.
5. Update  $\omega_i$  and  $\mu_i$ .
6. Evaluate  $\sigma_b^2(t)$ . Desired threshold corresponds to maximum  $\sigma_b^2(t)$  by eq. (14)

$$\sigma_\omega^2(t) = \omega_1(t) \sigma_0^2(t) + \omega_1(t) \sigma_1^2(t) \quad (14)$$

where  $\omega_0$  and  $\omega_1$  are variances of these two classes, and  $\sigma_0^2$  and  $\sigma_1^2$  are probability of two classes that are separated by threshold  $t$ . The  $L$  bins of the histogram are used to calculate the class probability of 0 and 1 ( $t$ ) by eq. (15-16).

$$\omega_L(t) = \sum_{t=0}^{t-1} p_{iin} \quad (15)$$

$$\omega_1(t) = \sum_{i=t}^{L-1} \sum_{ih} \quad (16)$$

Fuzzy demonstrates the similarity between maximising inter-class variance and inter-class variance by eq. (17).

$$\sigma_{\omega}^2(t) = \omega_1(t)\sigma_0^2(t) + \omega_1(t)\sigma_t^2(t) \quad (17)$$

which is defined in terms of the class mean  $\mu$  and class probability  $\omega$ . The class mean is  $\mu_0, 1, T(t)$  is given by eq. (18,19)

$$\mu(t) = \frac{\sum_{i=0}^{L-1} i(i)}{\omega(t)} \quad (18)$$

$$\mu_1(t) = \frac{\sum_{i=t}^{L-1} i(i)}{\omega_k(t)} \quad (19)$$

It is simple to verify the relationships listed by eq. (20,21):

$$\omega_b\mu_0 + \omega_1\mu_1 = \mu_T \quad (20)$$

$$\omega_0 + \omega_1 = 1 \quad (21)$$

The FDL stabilisation constant is discarded by the initial factor. The following factor will offer high member esteems. The third component represents enthusiasm. It provides the clustering process with more tractability and competency. It is stated what the GPD is by eq. (22):

$$P_c = \prod_{k=1}^E N(b_k | \mu_b, \Sigma_b) \quad (22)$$

Mean of dataset is symbolized by  $\mu_b$  eq. (23)

$$\mu_b = \frac{1}{g} \sum_{h=1}^g d_h \quad (23)$$

Data covariance is given by eq. (24)

$$\Sigma_b = \frac{1}{g} \sum_{h=1}^g (d_h - \mu_b) * (d_h - \mu_b)^T \quad (24)$$

Where  $g$  stands for the user-imposed restriction that interferes with the strong point. These factors are utilized to evaluate probabilities of the data as well as parameters together.

#### Algorithm-1 Conv\_FCMT:

```

INPUT : Color RGB image
OUTPUT : Segmented binary image
Scan the input RGB image - Image
Gimage ← Gray (Image)
Row ← No. of rows in Gimage
Col ← No. of columns in Gimage
Create ' k ' number of clusters [Clust, 0 < k ≤ 5 ]
∀ Cluster, Initialize Tot pixel no ok ← 0
for r = 1 to Row do

For j=1 to Col do

Update Tot nok if Gimage

End

End

Choose Clustk that has the max

Z=Tot pixel nok

Calculate the threshold value T such that

Segment and generate a binary image using T value

Simage =  $\begin{cases} 0 & \text{if Simage}(m, n) > T \\ 1 & \text{if Simage}(m, n) < T \end{cases}$ 

```

**Gradient multilayer Kernelized perceptron integrated with Darwinian optimization (GMKP-DO):**

Consider a d-dimensional pattern  $x$  from an input domain  $X$ , which is typically Euclidean vector space  $R^d$ , as input to network by eq. (25).

$$\theta_h = z(\langle w_h, x \rangle), h = 1, \dots, H. \quad (25)$$

The hidden layer's  $H$  input-to-hidden mappings  $\theta_h$  are combined into an  $H$ -dimensional feature vector,  $\theta = [\theta_1 \cdot \dots \cdot \theta_H]$ , which uses the notation  $\sigma_\omega^2(t) = \omega_1(t)\sigma_0^2(t) + \omega_1(t)\sigma_i^2(t)$ . Under this view, the training samples are transformed into information particles. The order  $\alpha$  potentials and information forces can be defined using the new nonparametric entropy estimator. The potential estimator for information is given by eq. (26)

$$\frac{\partial \hat{V}_\alpha(e)}{\partial w} = (\alpha - 1) \sum_j \hat{f}_e^{\alpha-2}(e_j) \frac{\partial \hat{V}_2(e_j)}{\partial w} \quad (26)$$

where the unit-size  $\sigma$  kernel is used to express the n-dimensional size-kernel as  $\kappa_\sigma(x) = \frac{1}{\sigma^n} \kappa'(x/\sigma)$ . An immediate deduction of the potential energy of a data particle,  $j$ , is made from (27,28).

$$\begin{aligned} \frac{\partial E^\Phi(\alpha, \beta)}{\partial \alpha_m} &= - \sum_{j=1}^l [y_j - f(\hat{y}_j)] f'(\hat{y}_j) k(\mathbf{x}_m, \mathbf{x}_j) \\ m &= 1, \dots, l \\ \frac{\partial E^\Phi(a, b)}{\partial \beta} &= - \sum_{j=1}^l [y_j - f(\hat{y}_j)] f'(\hat{y}_j) \end{aligned} \quad (27)$$

$$\text{where } \hat{y}_j(t) = \sum_{i=1}^l \alpha_i(t) k(\mathbf{x}_i, \mathbf{x}_j) + \beta(t) \quad (28)$$

The information force on  $j$  can be calculated from (29).

$$F_\alpha(e_j) = \frac{\partial \hat{V}_\alpha(e_j)}{\partial e_j} = \frac{(\alpha-1)}{N^\alpha} \left( \sum_i \kappa_\sigma(e_j - e_i) \right)^{\alpha-2} \left( \sum_{i=j} \kappa'_\sigma(e_j - e_i) \right) \quad (29)$$

This may be demonstrated to be equivalent to eq. (30)

$$F_\alpha(e_j) = (\alpha - 1) \hat{f}_e^{\alpha-2}(e_j) F_2(e_j) F_\alpha(e_j; e_i) = (\alpha - 1) \hat{f}_e^{\alpha-2}(e_j) F_2(e_j; e_i) \quad (30)$$

where the definition of the quadratic force is given by eq. (31)

$$F_2(e_j) = \frac{1}{N^2} \left( \sum_{i=j} \kappa'_\sigma(e_j - e_i) \right) F_2(e_j; e_i) = \frac{1}{N^2} \kappa'_\sigma(e_j - e_i) \quad (31)$$

This makes the connection between the quadratic force and the  $\alpha$ -force more clear. Evidently, when Gaussian kernels are taken into consideration, quadratic force expression in (14) reduces to same description in [Principe et al]. The force on  $j$  caused by  $i$  may now be defined from (32) and (33).

$$F_\alpha(e_j; e_i) = (\alpha - 1) \hat{f}_e^{\alpha-2}(e_j) F_2(e_j; e_i) \quad (32)$$

$$F_2(e_j; e_i) = \frac{1}{N^2} \kappa'_\sigma(e_j - e_i) \quad (33)$$

According to this definition, quadratic force is thought of as the root of all other informational forces. Force acting on a particle rises for  $\alpha > 2$  and falls for  $\alpha < 2$  as the probability density increases by eq. (34-36).

$$\alpha_m(0), \beta(0) : \text{arbitrary} \quad (34)$$

$$\begin{aligned} \alpha_m(t+1) &= \alpha_m(t) \\ &+ \rho_t \sum_{j=1}^l [y_j - f(\hat{y}_j(t))] f'(\hat{y}_j(t)) k(\mathbf{x}_m, \mathbf{x}_j) \end{aligned} \quad (35)$$

$$\beta(t+1) = \beta(t) + \rho_t \sum_{j=1}^l [y_j - f(\hat{y}_j(t))] f'(\hat{y}_j(t)) \quad (36)$$

Strong theoretical results support it and show how strong predictors may be created by iteratively combining weaker models using a greedy technique that is equivalent to gradient descent in a function space. A new gradient boosting decision tree approach called the CatBoost model properly handles categorical data and benefits from handling them during training rather than preprocessing time. The technique also reduces over-fitting because it chooses a fresh schema for computing leaf values while choosing tree structure. The CatBoost model employs a more effective method that minimises over-fitting and permits training on the entire dataset. Specifically, they randomly shuffle dataset, and then compute average label value for every sample with same category value shuffled in front of the provided one. When computing gradient of data potential, we come across information forces. The gradient is made up of the terms for data force as well as sensitivity, as shown below in (37).

$$\frac{\partial \hat{V}_\alpha(e)}{\partial w} = \sum_j \frac{\partial \hat{V}_\alpha(e_j)}{\partial e_j} \frac{\partial e_j}{\partial w} = \sum_j F_\alpha(e_j) S_w(e_j) \quad (37)$$

Gradient of data potential with regard to weights is explicitly expressed as eq. (38)

$$\frac{\partial \hat{V}_2(e_j)}{\partial w} \left[ \sum_i \kappa'_\sigma(e_j - e_i) \left( \frac{\partial y_i}{\partial w} - \frac{\partial y_j}{\partial w} \right) \right] \quad (38)$$

Where backpropagation can be used to calculate sensitivity of output for an MLP. Theoretically, it is interesting to investigate the connection between the quadratic data potential and gradient of order-data potential. Rearranging expression in (38) will result in eq. (39)

$$\frac{\partial \hat{V}_\alpha(e)}{\partial w} = (\alpha - 1) \sum_j \hat{f}_e^{\alpha-2}(e_j) \quad (39)$$

This characteristic directly relates to the information forces' observed properties. Gradient adaptation is popular because of its ease of use and effective convergence properties, albeit it is not the only option. A search is simply abandoned when it tends to a local optimum and a different area is searched in its place.

This method rewards swarms that show improvement at each stage as well as penalises swarms that remain stagnant. Optimization algorithm used for image segmentation is shown in Table I. Local best (xa[t]) and global best (ga[t]) data have a significant impact on the position and velocity data. When the new velocity is calculated, coefficients w, ρ1, and ρ2 are given weights that regulate inertial influence, i.e., according to "the globally best" and "the locally best," By giving each component a different set of influences; however, alternative outcomes might be achieved. Correctly tweaking these settings will produce better outcomes depending on the application and the peculiarities of the situation. Each element of the random vectors with the parameters r1 and r2 is typically a uniform random number between 0 and 1.

<b>Algorithm:</b>
<p>Start <math>\alpha, \rho_1, \rho_2</math>            Start <math>N, N_{\min}, N_{\max}</math></p> <p>Start <math>N^s, N_{\min}^s, N_{\max}^s</math>            Start <math>\Delta v</math></p> <p style="text-align: center;"><math>I_T, I_{kiu}</math></p> $p_i^c = \frac{\tilde{h}_i^c}{N}, \sum_{i=0}^{L-1} p_i^c = 1$ $\mu_T^c = \sum_{i=0}^{L-1} i p_i^c$ <p>Start <math>0 \leq x_a^s[0] \leq L - 1</math>            Start <math>\tilde{x}_a^s, \partial_a^s</math> based on <math>x_a^s[0]</math></p> <p>For every iteration <math>t</math> until <math>I_T</math></p> <p>For every particle swarm <math>a</math> of swarm <math>s</math></p> $v_a^s[t + 1] = \alpha v_a^s[t] + \frac{1}{2} \alpha v_a^s[t - 1] + \frac{1}{6} \alpha (1 - \alpha) v_a^s[t - 2] + \frac{1}{24} \alpha (1 - \alpha) (2 - \alpha) v_a^s[t - 3]$ $+ \rho_1 r_1 (g_a^s - x_a^s[t]) + \rho_2 r_2 (\tilde{x}_a^s - x_a^s[t]),  v_a^s[t + 1]  \leq \Delta v$ $x_a^s[t + 1] = x_a^s[t] + v_a^s[t + 1], 0 \leq x_a^s[t + 1] \leq L - 1$ <p>Evaluate (4) and (5) based on thresholds described in <math>x_a^s[t + 1]</math></p> $\sigma_B^{c^2} = \sum_{j=1}^n w_j^c (\mu_j^c - \mu_T^c)^2$ <p>If <math>\sigma_{aB}^{c^2} &gt; \sigma_{abest}^e \quad \frac{2}{g}/1</math> particle <math>a</math> has improved  <math>\sigma_{n\ best\ B}^c = \sigma_B^{c^2}</math>  <math>\tilde{x}_a^s = x_a^s[t + 1]</math></p> <p>For every swarm <math>s</math></p> <p>If <math>\max \sigma_{s=}^{c^2} &gt; \varphi^c //</math> swarm <math>s</math> has improved</p>

```

 $\varphi^c = \max \sigma_{s_b}^2$ 
 $\tilde{g}_a^s = x_a^s[t + 1]$ 
 $I_k = 0$  // reset stagnancy counter
If  $N_s < N_{\max}$  // number of allowed particles
 $N_s = N_s + 1$ 
If  $N^s < N_{\max}^s$  and  $\text{rand}() \frac{N_s}{N_{\max}} > \text{rand}()$ 
 $N^s = N^s + 1$ 
 $I_k = I_k + 1$ 
If  $I_k = I_{kia}$  // swarm  $s$  has improved for too long
If  $N_s > N_{\min}$ 
Delete worse particle from swarm  $s$ , i.e., lower local solution
Else Delete whole swarm  $s$ , all particles from swarm  $s$ 

```

#### 4. Performance Analysis:

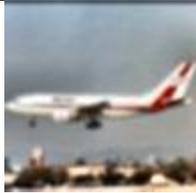
Implementation was carried out using the Python tool, and the following specifications were taken into account for the experiment: an Ubuntu-running PC, 4GB of RAM, and an Intel i3 processor.

1. Dataset description: ImageNet is one of the most popular datasets for the training of new algorithms, thanks to its sheer size and high variation. The dataset is organized in accordance with the application and around 3.5KB images are used to retrieve Content based images are used in the research. In total there

are more than one and a half million images. ImageNet constitutes the gold standard to which other image databases are compared to.

2. COREL- 10K: The Corel-10K datasets contains 10, 000 images. The groups are fruits, birds, globes, coins and many more. The image size is  $128 \times 187$  or  $187 \times 128$ . COREL-1K: The Corel-1K dataset contains 1000 images. The images are categorized into various groups such as buses, dinosaurs, flowers, beaches etc. Image size is  $256 \times 384$  otherwise  $384 \times 256$ .

**Table-1** Proposed technique based segmentation and classification for various images

Input image	Pre-processed image	Segmented image	Classified image
Flight			
Bird			
Car			

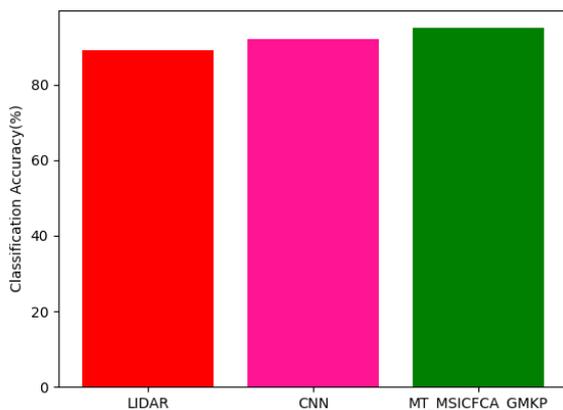
The above table-1 shows processing of various dataset based on segmentation and classification using proposed techniques. The input image of various dataset has been segmented and classified multispectral image is shown using proposed technique.

**Table-2** Comparative analysis between proposed and existing technique based on various **Input images**

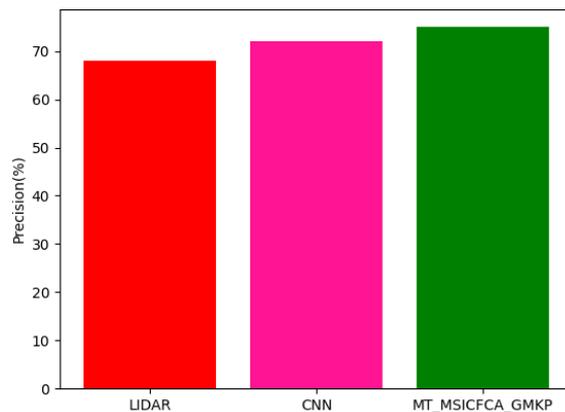
Input images	Techniques	Classification accuracy	Precision	Recall	F-Measure	ROC
Flight	LIDAR	89	68	55	42	32
	CNN	92	72	59	45	36
	MT_MSICFCA_GMKP	93	88	85	89	75
Bird	LIDAR	90	72	58	44	34
	CNN	92	75	62	48	36
	MT_MSICFCA_GMKP	94	92	90	90	41
Car	LIDAR	92	75	61	49	38
	CNN	94	79	63	53	42
	MT_MSICFCA_GMKP	96	82	65	56	45

The comparative comparison of suggested and current methodologies for various datasets is shown in table 2 above. In this case, parametric analysis is done in terms of classification precision, recall, F-measure, ROC. Projected DL method's general prediction capacity is used to calculate accuracy. For both the present and the proposed techniques, the number of pictures processed is the Input images signal when calculating F-score. Every feature's ability to differentiate from other features on its own is revealed by F-score. A score is generated for first feature, while a different score is derived for second feature. It makes no mention of how the two components interact, though. The prediction performance in this case was calculated utilizing exploitation to calculate the F-score. It is made by examining the harmonies of

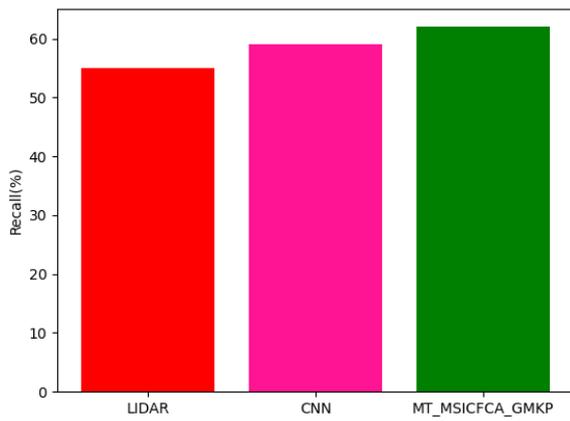
precision and memory. A calculated score of 1 denotes great performance, while a score of 0 denotes subpar performance. F-measures do not take into account the actual negative rate. By dividing total items categorized as belonging to a positive class by number of genuine positives, accuracy of a class is determined. Likelihood that a classification functions, if present, will generate a true positive rate. Recall is defined in this context as ratio of total number of components that actually belong to positive class to a number of true positives. Recall measures how well a method can identify Positive samples. RMSE is evaluated using the MSE squared root. The RMSE figures out how processing has altered each pixel. A statistic utilized to assess similarity between sample sets is Jaccard Index, commonly known as Jaccard similarity coefficient.



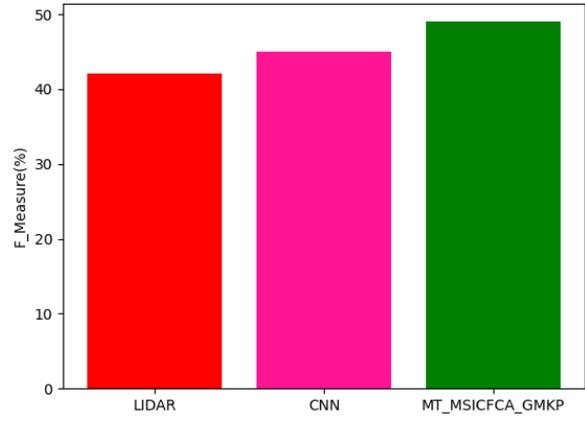
(a) classification accuracy



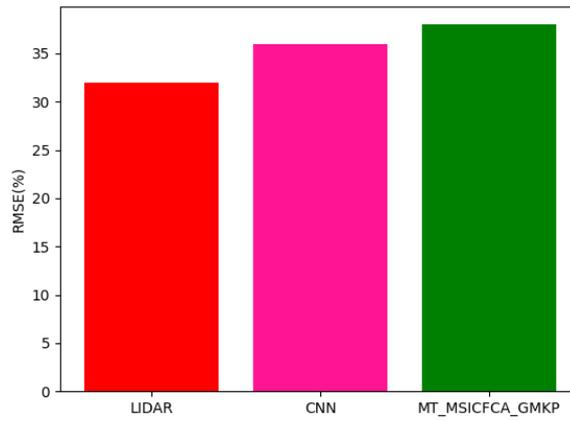
(b) precision



(c) Recall



(d) F-1 Score

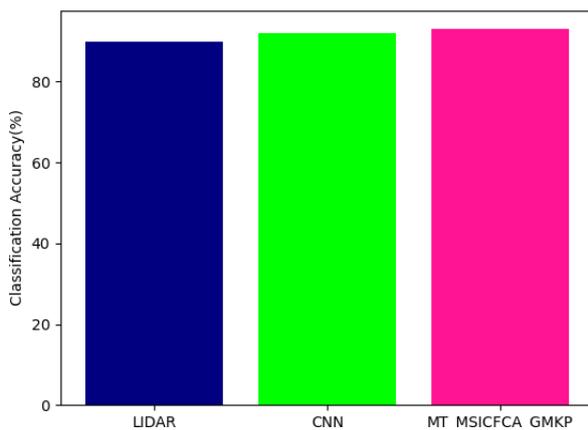


(e) ROC

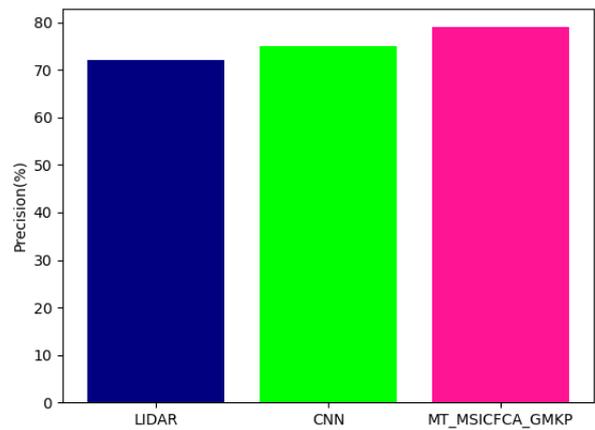
**Fig 3** Comparative analysis between proposed and existing technique for flight dataset in terms of (a) classification accuracy, (b) precision, (c) Recall, (d) F-1 Score, (e) ROC

Above figure-3 gives analysis between proposed and existing technique for flight dataset. Here proposed technique attained classification accuracy of 93%, precision of 88%, recall of 85%, F-measure of 89%, ROC of 75%, existing LIDAR attained classification

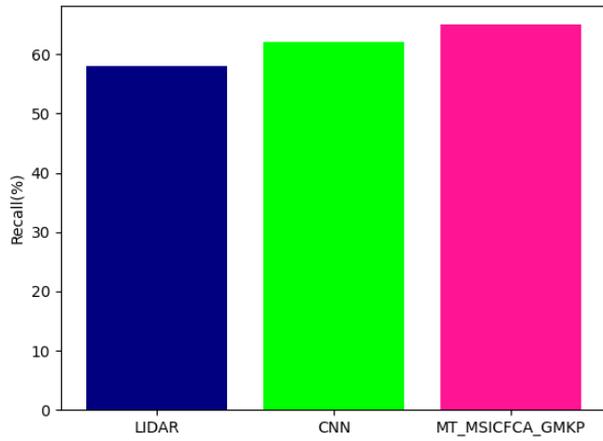
accuracy of 89%, precision of 68%, recall of 55%, F-measure of 42%, ROC of 32%; CNN attained classification accuracy of 92%, precision of 72%, recall of 59%, F-measure of 45%, ROC of 36%.



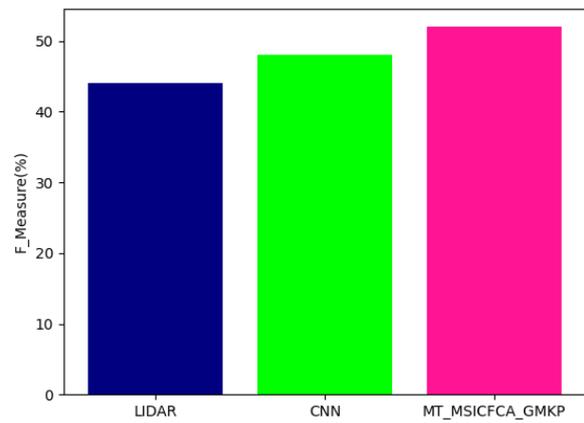
(a) classification accuracy



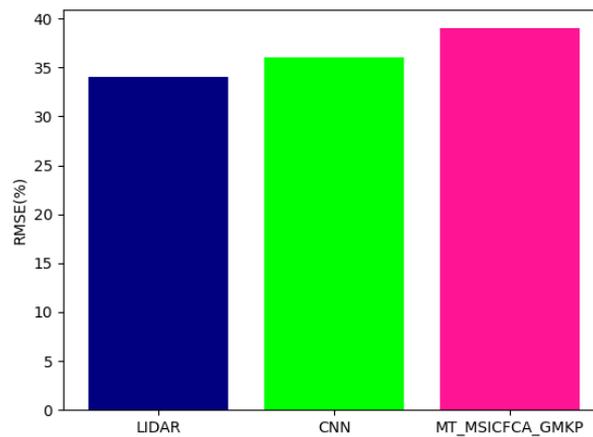
(b) precision



(c) Recall



(d) F-1 Score

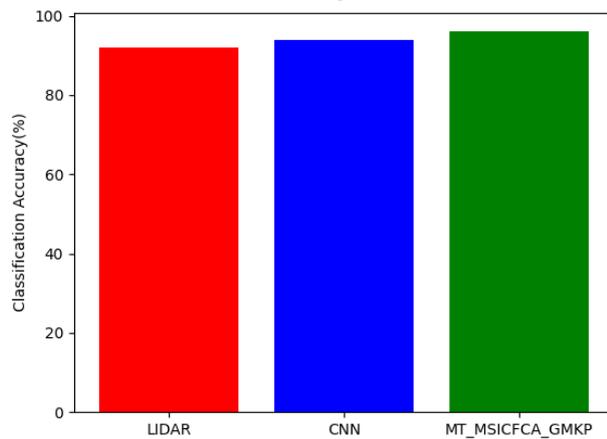


(e) ROC

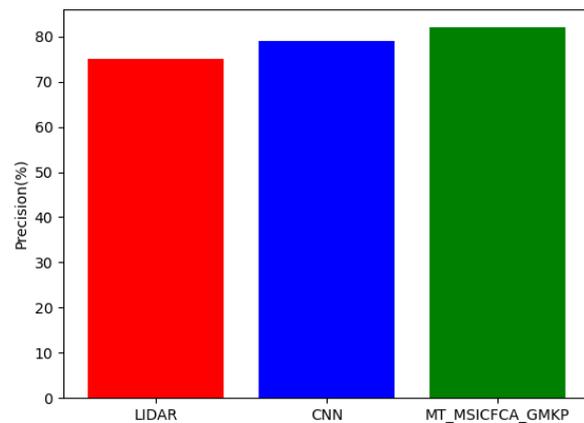
**Fig 4** Comparative analysis between proposed and existing technique for bird dataset in terms of (a) classification accuracy, (b) precision, (c) Recall, (d) F-1 Score, (e) ROC

From above figure-4 gives analysis between proposed and existing technique for bird image dataset. Here proposed technique attained classification accuracy of 94%, precision of 92%, recall of 90%, F-measure of 90%, ROC of 41%, existing LIDAR attained

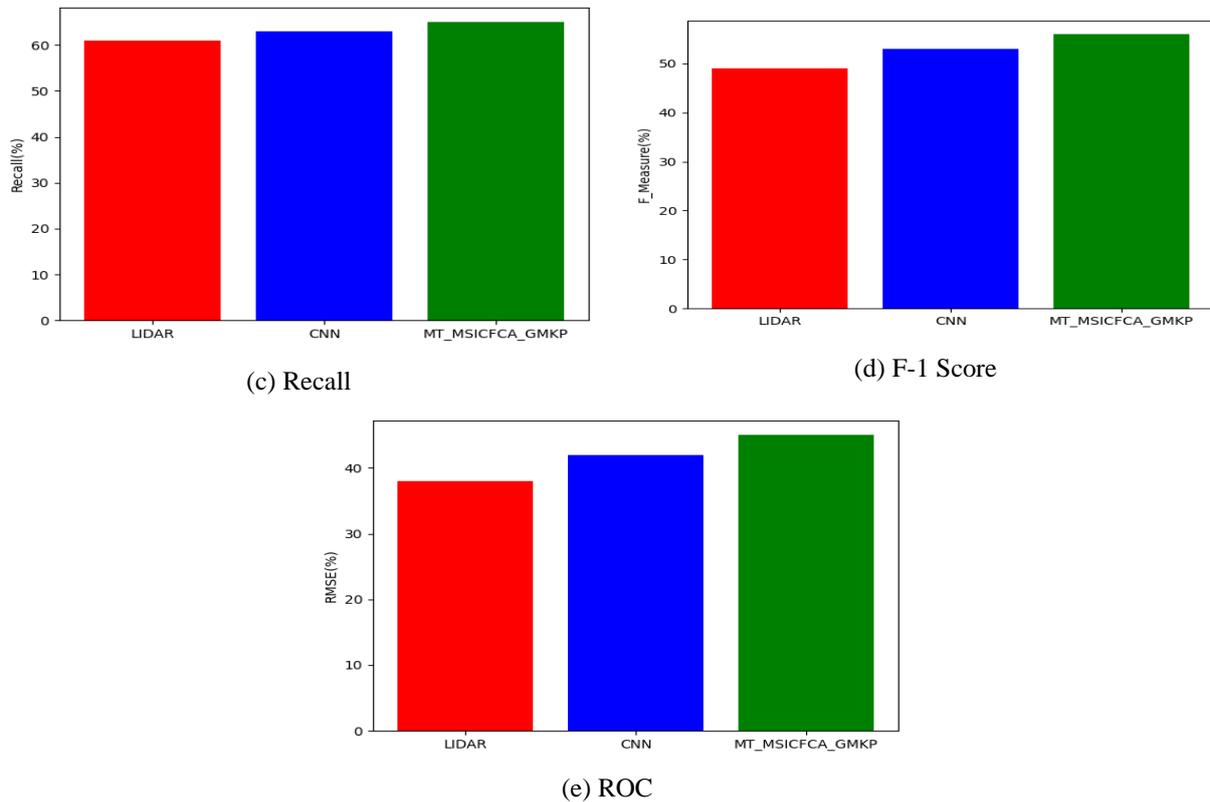
classification accuracy of 90%, precision of 72%, recall of 58%, F-measure of 44%, ROC of 34%; CNN attained classification accuracy of 92%, precision of 75%, recall of 62%, F-measure of 48%, ROC of 36%.



(a) classification accuracy



(b) precision



**Fig 5** Comparative analysis between proposed and existing technique for car dataset in terms of (a) classification accuracy, (b) precision, (c) Recall, (d) F-1 Score, (e) ROC

The above figure-5 gives analysis between proposed and existing technique for car dataset. Here proposed technique attained classification accuracy of 96%, precision of 82%, recall of 65%, F-measure of 56%, ROC of 45%, existing LIDAR attained classification accuracy of 92%, precision of 75%, recall of 61%, F-measure of 49%, ROC of 38%; CNN attained classification accuracy of 94%, precision of 79%, recall of 63%, F-measure of 53%, ROC of 42%.

## 5. Conclusion:

This research proposes novel method in various image analysis based on multilevel thresholding based on segmentation with classification using DL techniques. deep learning techniques significantly rely on availability of large amounts of high-quality data. Although visible light gives information that is nearly identical to what the human eye processes, it is occasionally unable to do so. Processed image is segmented using convolutional operation based fuzzy clustering with multilevel thresholding (Con\_Fuz\_Clus\_MT) and classified using gradient multilayer Kernelized perceptron integrated with Darwinian optimization (GMKP-DO). Proposed technique attained classification accuracy of 93%, precision of 88%, recall of 85%, F-measure of 89%, ROC of 75% for flight dataset; classification accuracy of 94%, precision of 92%, recall of 90%, F-measure of

90%, ROC of 41% for bird image dataset, classification accuracy of 96%, precision of 82%, recall of 65%, F-measure of 56%, ROC of 45% for car image dataset.

## References:

- [1] Henila, M., & Chithra, P. (2020). Segmentation using fuzzy cluster-based thresholding method for apple fruit sorting. *IET Image Processing*, 14(16), 4178-4187.
- [2] Raja, P. S. (2020). Brain tumor classification using a hybrid deep autoencoder with Bayesian fuzzy clustering-based segmentation approach. *Biocybernetics and Biomedical Engineering*, 40(1), 440-453.
- [3] Dixit, M., Chaurasia, K., & Mishra, V. K. (2021). Dilated-ResUnet: A novel deep learning architecture for building extraction from medium resolution multi-spectral satellite imagery. *Expert Systems with Applications*, 184, 115530.
- [4] Almahasneh, M., Paiement, A., Xie, X., & Abouadarham, J. (2022). MLMT-CNN for object detection and segmentation in multi-layer and multi-spectral images. *Machine Vision and Applications*, 33(1), 1-15.
- [5] Ravi, C., Yasmeen, Y., Masthan, K. ., Tulasi, R. ., Sriveni, D. ., & Shajahan, P. . (2023). A Novel Machine Learning Framework for Tracing Covid Contact Details by Using Time Series Locational

- data & Prediction Techniques. *International Journal on Recent and Innovation Trends in Computing and Communication*, 11(2s), 204–211. <https://doi.org/10.17762/ijritcc.v11i2s.6046>
- [6] Chouhan, A., Agrawal, A., & Sur, A. (2021, December). Unsupervised Change Detection in Very High Resolution Multi-Spectral Images. In *2021 IEEE International India Geoscience and Remote Sensing Symposium (InGARSS)* (pp. 293-296). IEEE.
- [7] Jena, B., Naik, M. K., Panda, R., & Abraham, A. (2021). Maximum 3D Tsallis entropy based multilevel thresholding of brain MR image using attacking Manta Ray foraging optimization. *Engineering Applications of Artificial Intelligence*, 103, 104293.
- [8] Wang, X., & Chen, Y. (2022). Research on Video Compression Algorithm Based on Deep Learning. In *International Conference on Computer Engineering and Networks* (pp. 1371-1378). Springer, Singapore.
- [9] Mr. Kaustubh Patil. (2013). Optimization of Classified Satellite Images using DWT and Fuzzy Logic. *International Journal of New Practices in Management and Engineering*, 2(02), 08 - 12. Retrieved from <http://ijnpme.org/index.php/IJNPME/article/view/15>
- [10] Maheswari, K. U., & Rajesh, S. (2022). Fused LISS IV Image Classification using Deep Convolution Neural Networks. *INTERNATIONAL JOURNAL OF COMPUTERS COMMUNICATIONS & CONTROL*, 17(5).
- [11] Cao, M. T., Nguyen, N. M., Chang, K. T., Tran, X. L., & Hoang, N. D. (2021). Automatic recognition of concrete spall using image processing and metaheuristic optimized LogitBoost classification tree. *Advances in Engineering Software*, 159, 103031.
- [12] Pal, R., Mukhopadhyay, S., Chakraborty, D., & Suganthan, P. N. (2022). Very high-resolution satellite image segmentation using variable-length multi-objective genetic clustering for multi-class change detection. *Journal of King Saud University-Computer and Information Sciences*.
- [13] Goyal, L. M., Mittal, M., Kumar, M., Kaur, B., Sharma, M., Verma, A., & Kaur, I. (2021). An efficient method of multicolor detection using global optimum thresholding for image analysis. *Multimedia Tools and Applications*, 80(12), 18969-18991.
- [14] Li, X., Ye, H., & Qiu, S. (2022). Cloud Contaminated Multispectral Remote Sensing Image Enhancement Algorithm Based on MobileNet. *Remote Sensing*, 14(19), 4815.
- [15] Navarro, P. J., Miller, L., Díaz-Galián, M. V., Gil-Navarro, A., Aguila, D. J., & Egea-Cortines, M. (2022). A novel ground truth multispectral image dataset with weight, anthocyanins, and Brix index measures of grape berries tested for its utility in machine learning pipelines. *GigaScience*, 11.
- [16] Vinaykumar, V., & Babu, J. A. (2022, July). Satellite Image Classification based on Adaptive Skip Connection-Convolutional Neural Network. In *2022 IEEE International Conference on Data Science and Information System (ICDSIS)* (pp. 1-4). IEEE.
- [17] Labh, J. R., & Dwivedi, R. K. (2021, December). Extensive Study on Color and Light Translation of 2D Images using Machine Learning Approaches. In *2021 10th International Conference on System Modeling & Advancement in Research Trends (SMART)* (pp. 290-294). IEEE.
- [18] Sangeetha, T., & Mohanapriya, M. (2022). A Novel Exploration of Plant Disease and Pest Detection Using Machine Learning and Deep Learning Algorithms. *Mathematical Statistician and Engineering Applications*, 71(4), 1399-1418.
- [19] Bharti, R., Saini, D., & Malik, R. (2021). A novel approach for hyper spectral images using transfer learning. In *IOP Conference Series: Materials Science and Engineering* (Vol. 1022, No. 1, p. 012120). IOP Publishing.
- [20] Ms. Elena Rosemaro. (2014). An Experimental Analysis Of Dependency On Automation And Management Skills. *International Journal of New Practices in Management and Engineering*, 3(01), 01 - 06. Retrieved from <http://ijnpme.org/index.php/IJNPME/article/view/25>
- [21] Dutta, T., Dey, S., Bhattacharyya, S., & Mukhopadhyay, S. (2021). Quantum fractional order darwinian particle swarm optimization for hyperspectral multi-level image thresholding. *Applied Soft Computing*, 113, 107976.
- [22] Oktem, F. S., Kar, O. F., Bezek, C. D., & Kamalabadi, F. (2021). High-resolution multi-spectral imaging with diffractive lenses and learned reconstruction. *IEEE Transactions on Computational Imaging*, 7, 489-504.
- [23] Popat, M., & Patel, S. (2022). Research perspective and review towards brain tumour segmentation and classification using different image modalities. *Computer Methods in Biomechanics and Biomedical Engineering: Imaging & Visualization*, 1-19.