

# A Novel Approach for Object Recognition Using Decision Tree Clustering by Incorporating Multi-Level BPNN Classifiers and Hybrid Texture Features

Upendra Kumar, Institute of Engineering and Technology, Dr. A.P.J. Abdul Kalam Technical University, India\*

 <https://orcid.org/0000-0003-3792-7945>

## ABSTRACT

This work proposes a novel approach to object recognition, particularly for human faces, based on the principle of human cognition. The suggested approach can handle a dataset or problem with a large number of classes for classification more effectively. The model for the facial recognition-based object detection system was constructed using a combination of decision tree clustering based multi-level Backpropagation neural network classifier-TFMLBPNN-DTC and hybrid texture feature (ILMFD+GLCM) and applied on NS and ORL databases. This model produced the classification accuracy ( $\pm$ standard deviation) of  $95.37 \pm 0.951877\%$  and  $90.83 \pm 1.374369\%$  for single input and  $96.58 \pm 0.5604582\%$  and  $91.50 \pm 2.850439\%$  for group-based decision for NS and ORL database respectively. The better classification results encourage its application to other object recognition and classification issues. This work's basic idea also makes it easier to improve classification management for a wide range of classes.

## KEYWORDS

BPNN, Context Window Based Texture of Pixels, Gray-Level Co-Occurrence Matrix, Human Cognition, Intensity Level Based Multifractal Dimension, Multi-level Backpropagation Neural Network

## 1. INTRODUCTION

A fundamental area of study in computer vision, deep learning, artificial intelligence, etc. is object detection. More difficult computer vision tasks, like target tracking, event detection, behavior analysis, and scene semantic understanding, require it as a necessary prerequisite. In order to create an effective object recognition system—in this case, face recognition—an attempt was made to integrate the human cognition principle. Numerous researchers have utilized decision tree-based classification approaches in a variety of fields, such as facial expression recognition (Salmam et al., 2016), group-based study to identify localized melanoma patients (Tsai et al., 2007), protein classification (Pepik et al., 2009), occlusions detection (Karthigayani & Sridhar, 2014), rule extraction for security analysis (Ren et

DOI: 10.4018/IJIRR.338394

\*Corresponding Author

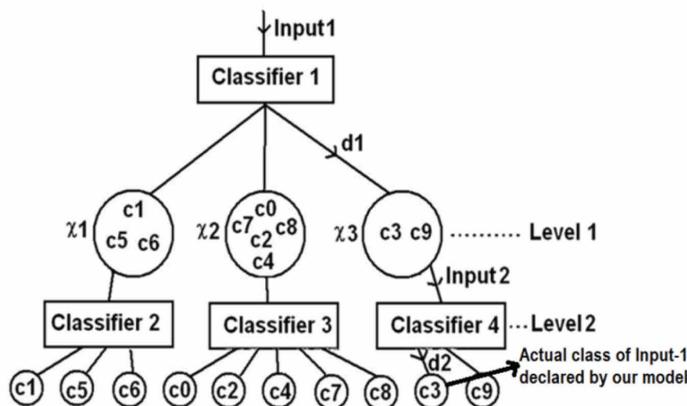
This article published as an Open Access article distributed under the terms of the Creative Commons Attribution License (<http://creativecommons.org/licenses/by/4.0/>) which permits unrestricted use, distribution, and production in any medium, provided the author of the original work and original publication source are properly credited.

al., 2006), gender classification (Khan et al., 2013), and hybrid classification based on decision tree and naïve bays methods (Muqasqas et al., 2014). In order to determine the most accurate class description for a given object, the decision tree concept takes into account the hierarchy of classes, just like humans do. This hierarchy starts at the top level of super-class hierarchy and moves down to the final hierarchy. However, there are very few classes at the base of the hierarchy, which makes it difficult for any classifier, including a human one, to handle due to its large number; as a result, the classes at this hierarchy must be grouped into a few clusters. For the classifier used to classify data into these super classes at this hierarchy, each of these clusters of classes can be thought of as a newly defined super class at the previous level of hierarchy. Similar methods have been applied to optical character recognition (Wilson et al., 1996) and face image recognition (Ebrahimpour et al., 2005) to increase classification accuracy. Using Fig. 1, the explanation that follows will help you understand this better. The majority of decision tree building techniques are based on decision clustering. Typically, decision trees are designed to handle a large number of classes. This method, of course, meets the needs of the majority of classifiers, most of which are unable to handle tasks involving a large number of classes. Figure 1 depicts a fictitious example of ten real classes, let's say  $c_1, c_2, c_3, \dots, c_{10}$ . In this case, at level 1 the original classes are merged to form super class's  $\chi_1, \chi_2$  and  $\chi_3$  for which classifier 1 is employed to get decision,  $d_1$  for class  $\chi_3$  at level 1. Next, for this decision as input, classifier 4 is invoked to reach the final decision  $d_2$  for class  $c_3$ . The flow of decision is as follows:

In order to mathematically adopt the object recognition system based on temporal description of the object, another cognition analog was also developed as (10). They contend that “we are continuously associating views of objects to support recognition, and that we do so not only on the basis of the objects’ correlated appearance in time, but also on their physical similarity.” Another argument in favor of this is that research by Bruce et al. (1998) and Knight & Johnston (1997) has shown and demonstrated that face recognition based on video is preferable than using still images. Additionally, they discover that motion aids in the recognition of (familiar) faces more effectively when the images are negated, or appropriately thresholded. This concept inspired the creation of moving head posed videos, which were recorded from left to right and vice versa, in order to extract multiple facial images of the same person in various poses and create a database for the classification model created in this work. The Bayesian Probabilistic model, which replicates the repeated input feature and aids in identifying the object with a higher probability, also supports this idea.

Figure 1. The process of making decision using “decision clustering tree” (common approach): Here input-1 is classified by root level classifier-1 to superclass- $\chi_3$ , where classifier-4 is employed, which classifies its actual class: class-3

Input1 → Classifier 1 →  $d_1$  →  $c_3$  → Input2 →  $d_2$  → Classifier 4 →  $c_3$



In this work, the classification was done using Backpropagation Neural Network (BPNN), a supervised learning-based technique. One unique kind of ANN model is the BPNN. Artificial neural network-based classifiers, or ANNs, are frequently used for object detection and recognition in a range of digital image processing applications. This is due to ANNs' superior efficiency, adaptability, robustness, resilience, and high accuracy performance, as well as their capacity to handle extremely non-linear and discontinuous classification problems (Jiang et al., 2010). In order to create classification models for use in image processing applications, researchers have built a range of ANN types with diverse network architectures in response to specific needs. Consequently, in order to train the ANN architectures, distinct types of learning algorithms are needed. Multi-layer perceptron (MLP) networks combined with back-propagation (BP) based training and learning algorithms are the most popular and common type of artificial neural network architecture (ANN) used in digital image classification tasks (Atkinson & Tatnall, 1997). The back propagation neural network (BPNN) is the name given to this kind of classification network. There are currently ongoing studies and in-depth investigations into how well BPNNs perform in handling classification problems involving image analysis. The study conducted by (Alsmadi et al., 2009) concludes that the back propagation (BP) learning algorithm has produced higher classification accuracy for a variety of applications and has been proven to be the best algorithm among all other types of multi-layer perceptron algorithms. However, the architecture and implementation of an ANN model's network often determines its processing speed, performance, and classification accuracy. Due to the BPNN classifier's successful applications, an automated system for the detection and recognition of facial images has been developed in this work.

Given the size of the database—40 facial images per person from a total of 117 subjects, or 4680 images—a hybrid approach that combines BPNN-based classification models with Decision Tree Clustering (DTC) was employed in this work to facilitate processing. Optical character recognition (Wilson et al., 1996), face image recognition (Ebrahimpour-Komleh et al., 2001), and data classification (Kaewchinporn, 2011) are just a few of the research projects that have employed the decision tree clustering approach to increase classification efficiency. Hierarchical clustering was used to classify the entire database into several super-classes. Following that, each of these distinct super-classes was treated as a single database for additional processing (a BPNN classifier was employed for classification in each super-class individually). The final classification results were then obtained by combining the classification results of all super-classes into a single group. When a single classifier is unable to classify a large dataset, this idea can be applied by splitting the dataset into smaller datasets for quicker processing. Despite having a limited set of data and classes, this protocol was also tested on a standard ORL database for comparative analysis.

Intensity level based multi fractal dimension (ILMFD), another cognition-inspired feature extraction technique, was also used in this work (the methodology to compute ILMFD is given in section 2.3.2.1). Fractal dimension (FD) was a tool that many researchers used to extract features from images for a variety of image processing applications (Kumar & Lahiri, 2013b; Tripathi et al., 2022). A FD is a pattern index or measure (in the form of integers or fractions) that is specifically used to count the complexity of objects' fractal patterns or feature sets. Any object's complexity can be expressed as a ratio of the detail change in the pattern to the dimension or scale change. The science of fractal analysis essentially rests on scaling rules. This scaling rule serves as an example of clearly stated conventional geometry and dimension rules. For digital images, FD is an essential metric for measuring and characterizing self-similarity and roughness (Chaudhuri & Sarkar, 1995). It can identify a wide range of picture classes and effectively describe the intrinsic information of the objects, something that other types of texture analysis methods are unable to accomplish. It was suggested that ILMFD be used to classify and identify real-world objects and images when the ROI requires intricate analysis (Kumar & Lahiri, 2013b). Their findings suggested that it was a superior choice over the conventional intensity-based bit-plane slicing (n-bit slicing). Eqn. 1 allows for the following expression of it:

$$ILMFD = \sum_{i=1}^n ILFD_i \quad (1)$$

where, ILMFD is the collection of n-dimensional multi-fractal dimensions that are determined from each sliced binary image, depending on the intensity level. When a digital image I has a maximum intensity value of  $I_m$ , the intensity value of the i-th slice image will have a range defined by Eqn. 2 as follows for a number of slices n:

$$(i-1) \frac{I_m}{n} \leq I_i < i \frac{I_m}{n}, i=1, 2, 3, \dots, n \quad (2)$$

where  $I_m$  is the highest intensity value of the specified original image I and  $I_i$  is the intensity value of the i-th slice. Unlike the traditional approach, which divides bit planes into equal intensity ranges, this approach achieved the same result by slicing the images into 10 equal intensity planes based on the criteria listed in Table 2, taking into account that  $I_m$  is the maximum value intensity value present in the image. Figure 5 appears to demonstrate the experimental fact of computing feature sets for improved analysis using different image slices.

FD is a crucial fractal geometry measurement parameter that has demonstrated broad and widespread applications in a number of fields, including digital image processing, particularly in the field of medicine. In order to develop a recognition system, image analysis is a high-level image processing technique that requires first extracting features from the image, such as texture, blobs, roughness, region of interest (ROI), smoothness, surface area, and the solidity of the objects. In their 2012 study, Shanmugavadivu & Sivakumar (2012) presented a method for determining the feature distance (FD) of a range of digital images. They then compared the FD to all other feature metrics for the same images and came to the conclusion that FD is the best feature extraction tool for precisely measuring the texture and roughness of object images. Numerous applications based on pattern recognition and digital image processing have effectively used the FD. Fractal geometry is a useful tool for estimating the texture-based characteristics of objects with complex figures, irregular shapes, and hidden patterns, according to studies by Keller et al. (1989) and Ranganath & Mishra (2017). By generating biological structures, it can effectively be applied to classification problems based on texture. In order to perform quantitative analysis of objects based on their shapes and structures, it essentially derives parameters known as FDs. Working in the crucial field of structural analysis, particularly in nanoscience and nanotechnology, was Ghaderi (Ghaderi, 2023). In this work, fractal data analysis was done using digital images of the nanocomposite obtained from scanning electron microscopy (SEM). To ensure the homogeneity and uniformity of SEM images, a range of characteristics of nanocomposite materials, including size, texture, dimensions, shape, and morphology, were examined by computing metrics of fractal dimensions. Fractal data analysis has emerged as a superior option for structural analysis of other types of images for future research, particularly for healthcare applications like medical-based MRI images, based on the notable improvement in results. Pixel range calculation (PRC) method, which is based on a modified version of the box counting method, was used by Ranganath et al. (2021) for fractal analysis of digital images to determine their texture, smoothness, or roughness.

Applications for face recognition have cited a few instances of fractal-based image coding (Tan & Yan, 199; Hossein et al., 2001; Kouzani & Sammut, 1997). Fractals have been effectively applied to the measurement of generalized information content in the gray level in texture classification (Ferens & Kinsner, 1995). There is evidence in (Kouzani & Sammut, 1997; Deaton et al., 2002; Cheng & Yueh-

Min, 2003) regarding the direct application of fractal dimension in image classification on facial, brain MRI, and retinal images. By definition, fractals are able to capture general pattern information. It was determined to use the intensity-level Fractal Information of each sliced-plane of an image as a logical feature of the corresponding image because face images are thought to convey both spatial and depth information. To obtain a set of 10-dimensional feature vectors, each image was sliced into ten equal intensity range planes using a heuristic method. This method's classification result has demonstrated the model's solid support in this work. Duolin & Wei (2022) also carried out work based on deep learning and fractal theory for image detection and classification. They suggested a deep learning and fractal feature fusion biometric authentication and monitoring system. Similar to this, Bisogni et al. (2020) proposed a model to estimate head pose using fractal coding theory, specifically using a metric based on Partitioned Iterated Function Systems (PIFS) to extract the fractal feature from the head pose images. This representation was then effectively compared to the fractal feature-based code of a reference model approaches using the metric of hamming distance. In a different study by Abdullahi et al. (2020), machine learning techniques are combined with hashing to create a reliable and alignment-free classification system for fingerprint image data. These academic works served as inspiration for the creation of an object recognition model that employed fractal features and machine learning techniques.

In order to prepare hybrid feature, other texture feature Gray-level Co-occurrence Matrix (GLCM) was explored in this work. Many researchers have used GLCM as a texture feature in facial recognition process (Salman et al., 2022; Vera et al., 2023; Alazawi et al. 2019). Finally hybrid feature was prepared by combing ILMFD+GLCM in order to train the model effectively. In the end, our suggested model significantly increased the classification efficiency of face recognition for both the standard ORL database and the non-standard database that we created.

This article's overall content is divided into the following sections:

- The primary contributions made by other researchers to the field of object detection and recognition are covered in Section 1, along with the use of models inspired by cognition for feature extraction and classification in a range of object detection and recognition contexts.
- The theoretical background of the procedure for gathering facial image data from ORL and non-standard databases (NS) are covered in Section 2. Following that, a detailed discussion with an appropriate diagram was given of pre-processing, feature extraction method (ILMFD and GLCM), multi-level BPNN classification method, and decision tree clustering approach.
- The experimental results and discussion are shown in Section 3, with a primary focus on the use of decision clustering and superclass-sub-class based multi-level BPNN classification to achieve classification efficiency for both NS and ORL databases.
- The conclusion and future scope of this work are covered in Section 4.
- The limitations and potential applications of the suggested model are covered in Section 5.

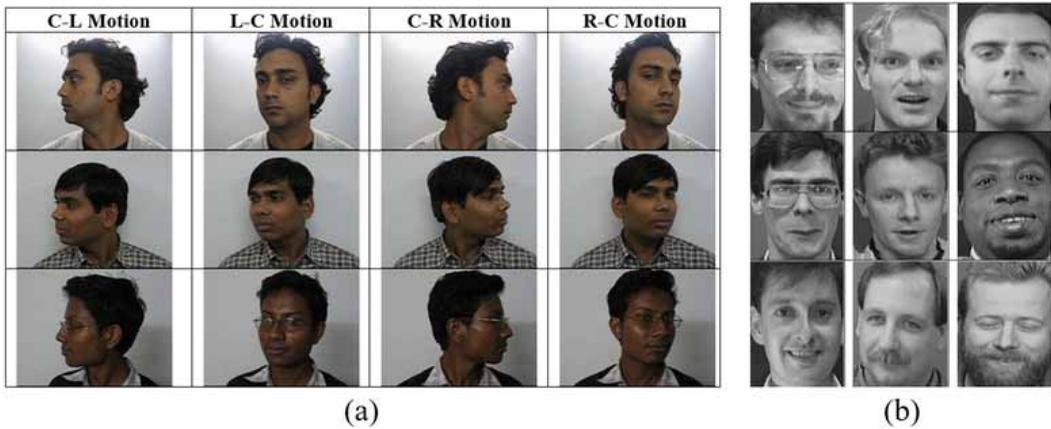
## 2. COMPUTATIONAL METHODOLOGY

The procedure for gathering facial image data from ORL and non-standard databases (NS) has been covered in this section. After that, a detailed discussion with an appropriate diagram covered pre-processing, facial segmentation using CWTP & BPNN based model, ILMFD based feature extraction method, multi-level BPNN and Decision Tree Clustering (TFMLBPNN-DTC) based classification, and ultimately decision clustering approach for group based decision based outcome.

### 2.1 Collection of Facial Image Frames

The facial images of variety of people (samples displayed in Fig. 2) were captured using a digital camera (Canon: Power Shot S50). As stated below, our database consists of two distinct types of databases:

Figure 2. Sample facial image frames used for classification work to build the model (a) in RGB format from our own NS database with different orientation from right to left ward (b) in Gray scale format from standard ORL database



- a) The Non-Standard Database (NS) contains 117 distinct subjects, each of which has four face-motion video sessions. Each person's face movements are as follows:
  - i) From the center, or frontal portrait position, to the right (C-R motion).
  - ii) R-C motion: moving from a right position toward the center.
  - iii) Moving leftward (C-L motion) from the center, or frontal portrait position.
  - iv) Moving in an L-C motion, from the left position to the center.
- b) A public database (the 40 unique subjects in the ORL Face Database (36), each with 10 image frames). Some people's facial images in this database were taken at different times, under different lighting conditions, with their faces in a variety of expressions (open or closed, smiling or not smiling), as well as with different external details (like glasses or not).

## 2.2 Data Preprocessing

We extracted image frames (40 images per person) at a constant sampling rate from video sessions. From the four face-motion clips listed above, 10 frames were taken at a constant sampling rate from each motion video clip of a person, resulting in a total of 40 frames per person across the four video sessions. The quality of the facial image frames was improved and prepared in a way that would facilitate further processing and yield useful results (Gonzales & Woods, 2002):

To obtain low dimensional data for easier processing, all face image frames were converted from color to grayscale.

- i) Face segmentation, or the extraction of the facial portion from each image frame, using BPNN as the classifier model and context window based texture of pixels (CWTP) as a feature vector (Kumar & Lahiri, 2013a).
- ii) After the facial segmentation process in step (ii), intensity-plane slicing (10 equal intensity range sliced planes) was completed to obtain intensity-slices of each image frame (discussed in methodology section).

## 2.3 Extracting Texture Feature From Sliced Planes of Each Image Frame

According to Schmidhuber, J. (1998), "The image of a female face with high ratings is composed from a fractal geometry based on rotated squares and powers of 2". This assertion also served as inspiration for the use of fractal dimension (Mandelbrot (1983) to extract better features from the images, as

well as multifractal dimension (ILMFD) for face recognition and pattern recognition from moving face sequences. Along with ILMFD, GLCM was also computed to build the hybrid texture feature.

### 2.3.1 Our Proposition for Cognition Based Texture Feature Extraction Method

Some of the first researchers to study learning in cortical circuits recognized the notion that temporal information could be used to establish spatial representations. The two findings that Wallis and Bulthoff (2001) produced after a thorough investigation into temporal association in recognition served as the cornerstone for our work. They are as follows:

- i) The temporal description of the object of interest is how human cognition occurs.
- ii) Cognition is a very quick process.

The initial assumption for the adoption of the temporal description of the object of interest was that, over a short period of time, the object description at any given instant is comparable to that at any other instant. One possible method to quantify this self-similarity mathematically is to find their fractal parameter. Furthermore, by making this assumption, we can measure this fractal parameter across a small number of discrete time scales, which merely expedites the recognition process. We built our object recognition system using these two unique powers of the fractal parameter. Also, in view of wide applications of GLCM in image classification problems (Salman et al., 2022; Vera et al., 2023; Alazawi et al. 2019), GLCM feature was also computed to build the hybrid texture feature.

### 2.3.2 Methodology for Texture Feature Extraction for Both ILMFD and GLCM

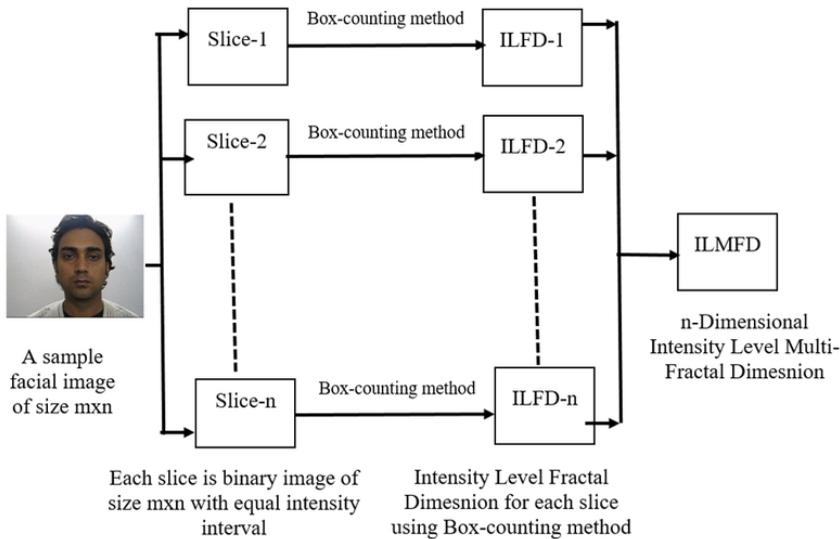
By calculating the fractal dimension for each sliced image (from the set of 10 sliced image planes obtained, as discussed in the section below), a multi-fractal dimension feature was obtained using the Box-counting method (Mandelbrot (1983)). An image's intensity slicing technique is covered in more detail in the following section-2.3.2.1. For every facial image, the collection of these fractal dimensions creates a feature vector known as the intensity level based multi fractal dimension (ILMFD). Following that, each facial image's GLCM feature was calculated using  $13 \times 4 = 52$  feature attributes and four directions ( $0^\circ$ ,  $45^\circ$ ,  $90^\circ$ ,  $135^\circ$ ). The hybrid feature of ILMFD+GLCM ( $10+52=62$  Dimensional) was fed to the TFMLBPNN-DTC classifier model to classify the data in their actual classes.

#### 2.3.2.1 Overview of Formulated Protocol (ILMFD Feature)

By slicing the face photos using an equal intensity interval technique, the feature was computed. The diagram provided in Fig. 3 can also be used to explain the steps involved in formulating an ILMFD feature. These steps can also be broken down into the following steps:

- Step 1: The frame extraction software received motion video samples or clips as input.
- Step 2: From the motion video clips above, facial image frames were extracted at a constant sampling rate.
- Step 3: Repeat Steps 4 through 6 for every frame.
- Step 4: To obtain the region of interest—in our case, the face—as a binary image, image enhancement and segmentation were performed (segmentation technique adapted from (37)).
- Step 5: As shown in Table 2, each image frame was divided into ten slices with fixed intensity intervals.
- Step 6: Using the methodology described in section 2.3.2.5, determine the fractal dimension (FD) for each intensity-sliced image (ILFD).
- Step 7: A collection of ILFDs taken from the ten fixed intensity interval-planes of every frame mentioned above yields an ILMFD, or multifractal dimension. Every ILMFD is arranged in a  $m \times n$  two-dimensional array, where  $m$  denotes the quantity of extracted frames and  $n$  the quantity of bit-slices.
- Step 8: The ILMFD feature was used to process face recognition data further.

Figure 3. A flow diagram depicting the extraction of feature, ILMFD for image frames of a person in which frames are extracted from video stream and further sliced into ten equal intensity interval planes, then fractal dimension is calculated for each slice (ILFD) and finally all ILFDs are combined to form ILMFD (Kumar & Lahiri, 2013b)



### 2.3.2.2 Slicing Images Into Equal Intensity-Range Planes and Multifractal Concept

The multifractal concept deals with objects that require more than one fractal dimension to describe them, as they cannot be adequately described by a single fractal dimension. A Multifractal feature or spectrum, as opposed to a single fractal dimension, can describe an object more effectively because of its multiple fractal dimensions. The study of a physical or other quantity's distribution on a geometric support is its main focus (Xie et al., 1998). This support may take the form of a fractal, a plane, the surface of a sphere, or a volume (Feder, 1989; Li et al., 1991). It can effectively handle objects with varying fractal dimensions in various locations. It has been effectively applied to recognize the texture of finger prints from pictures (Singh et al., 2005).

This work introduced the idea of fractals at different intensity planes, also known as the grey level plane, and used binary images of these planes to derive a set of fractal dimensions. The technique presented in this work is especially useful for features extraction from an image that are independent of scale and rotation. Based on the range of intensity values displayed in Table 1, multifractal dimension  $f_D$  is typically regarded as a collection of eight fractal dimensions (Singh et al, 2005). It has the following definition (Eqn. 3):

$$f_D = \sum_{D_i=0}^8 f_{D_i} \quad (3)$$

whereas in this work, 10 intensity sliced planes of equal intensity intervals were used in place of 8 bit-sliced binary images,  $f_D$  is a set of 8 fractal dimensions, each of which is measured from each of the 8 bit-sliced binary images obtained by intensity-slicing method. Each image is divided into eight intensity planes using the conventional bit slicing method, as shown in Table 1 (the gray scale range is 0-255). One of the method's limitations is that bit planes aren't separated into equal intensity ranges to capture the texture of different regions. In order to get around this, the images in this work were divided into ten equal intensity planes using the standards listed in Table 2 and Fig. 4. A set of 10 fractal dimensions (as ILMFD feature) was formulated as follows Eqn. 4:

Table 1. In the classic bit plane slicing method, the intensity ranges of 8-bit planes (gray scale intensity range: 0-255)

Bit Plane	Lower Intensity Value	Upper Intensity Value
1	0	2
2	> 2	4
3	> 4	8
4	> 8	16
5	> 16	32
6	> 32	64
7	> 64	128
8	> 128	255

Table 2. 10 Intensity planes with distribution of their corresponding intensity ranges with fixed interval  $\sim I_{max}/10$

Intensity Plane	Lower Intensity Value	Upper Intensity Value	Interval length
1	0	$I_{max}/10$	$I_{max}/10$
2	$> I_{max}/10$	$2I_{max}/10$	
3	$> 2I_{max}/10$	$3I_{max}/10$	
4	$> 3I_{max}/10$	$4I_{max}/10$	
5	$> 4I_{max}/10$	$5I_{max}/10$	
6	$> 5I_{max}/10$	$6I_{max}/10$	
7	$> 6I_{max}/10$	$7I_{max}/10$	
8	$> 7I_{max}/10$	$8I_{max}/10$	
9	$> 8I_{max}/10$	$9I_{max}/10$	
10	$> 9I_{max}/10$	$I_{max}$	

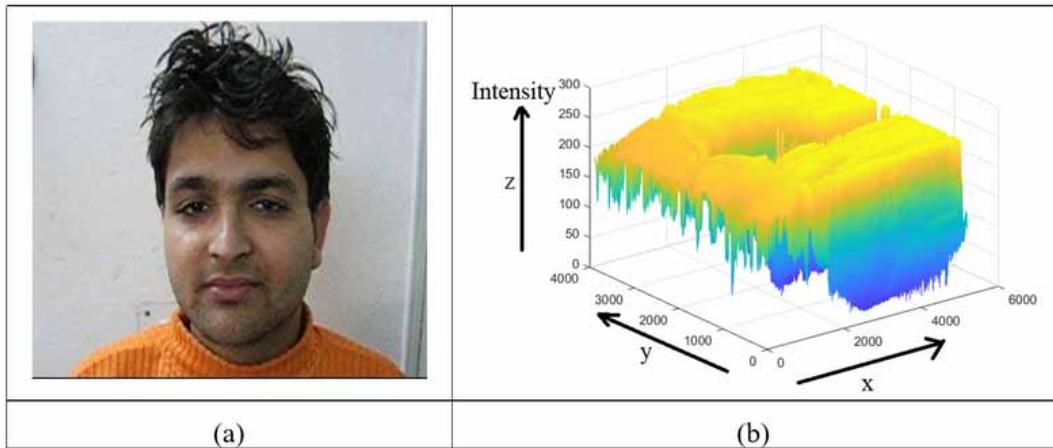
$$f_D = \sum_{D_i=0}^{10} f_{D_i} \tag{4}$$

where, each fractal dimension is determined from each of the ten binary images, and  $f_D$  is a set of ten fractal dimensions. Every sliced image out of ten would have an intensity value within the range shown in Table 2 if  $I_{max}$  is the maximum intensity value for any image. Thus, instead of using eight bit-sliced planes to form the feature vector in this work, ten intensity sliced planes of fixed-interval were used. The extracted image intensity sliced frames were utilized to compute the area-based fractal dimension, which ultimately formed the feature vector. This vector was thought to include the spatial details of every individual's facial image. Fig. 4 (which displays the distribution of intensity values in the various textures present in the image in three dimensions) provides a clear understanding of the process of slicing a facial image into n number of sliced image frames of fixed intensity intervals. The results obtained in this work concludes that this approach produced better results than the 8-bit slicing method.

### 2.3.2.3 Mathematical Formalism for Slicing Images

For a give facial image  $I$ , any of its  $i$ -th intensity plane,  $I_i$  is basically a binary description of  $I$  is defined as per the following rule (Eqn. 5):

Figure 4. 3-D distribution of intensity values of a facial image supports to slice the image into 'n' number of slices to compute ILMFD (a) a sample facial image and (b) its 3-D intensity plot to explain its slicing into 'n' intensity planes of equal interval against distribution of intensity values



$$\left. \begin{array}{l} \text{If } I_{\max} \geq I(\text{row}, \text{col}) \geq I_{\min} \\ \quad I_j(\text{row}, \text{col}) = 1; \\ \text{else} \\ \quad I_j(\text{row}, \text{col}) = 0; \end{array} \right\} \quad (5)$$

where  $I_{\min}$  and  $I_{\max}$  represent the lowest and highest intensities of the intensity range that the sub-object (in this case, a portion of a face) is embedded in. Therefore, we thought about using the following streamlined guidelines for the box counting method on binary images:

- The side-length of these boxes is an integer multiple of the binary image's smallest box, or pixel.
- The background and objects, or faces, have pixel values of 0 and 1, respectively.
- The box values that are chosen for counting contain any pixel with a value of 1.

### 2.3.2.5 Measuring FD Using Box Counting Method

Using the box counting method, the fractal dimension was calculated for each intensity plane (Mandelbrot (1983)). The simplest way to determine the fractal dimensions of an object with a grey value of 1 inside a binary image with a background of zero is to use the box counting method. It says that,

$$N(r) \sim r^{-D} \quad (6)$$

where  $D$  is the area-based fractal dimension of the objects, and  $r$  is the side-length of the square box,  $N(r)$  is the number of boxes needed to cover the measure of the object (number of pixels covering the measure, area of the object, in our case). As a result, a 10-dimensional feature vector known as ILMFD is created by adding the 10 fractal dimensions that correspond to the 10 sliced planes of each image (as indicated in Table 3).

**Table 3. A sample set of 10 fractal dimensions (columns) corresponding to ten intensity-sliced planes for five people (rows) separately**

Image No	Intensity plain 1	Intensity plain 2	Intensity plain 3	Intensity plain 4	Intensity plain 5	Intensity plain 6	Intensity plain 7	Intensity plain 8	Intensity plain 9	Intensity plain 10
1	1.9245	1.7225	1.6031	1.5784	1.6494	1.6192	1.3913	1.4429	1.4124	1.6266
2	1.9266	1.7297	1.6377	1.5901	1.6331	1.6191	1.395	1.4542	1.4352	1.5273
3	1.9219	1.7274	1.6368	1.5925	1.6303	1.6232	1.3726	1.3933	1.4082	1.5301
4	1.9278	1.7533	1.6434	1.5961	1.6324	1.5984	1.4068	1.4905	1.4132	1.5383
5	1.9289	1.7275	1.6107	1.5723	1.6514	1.628	1.3753	1.4383	1.3435	1.5329

### 2.3.2.6 Overview of Formulated Protocol for GLCM Feature

GLCM is a second-order statistical texture analysis method. By examining their spatial relationship, it defines the frequency of a combination of pixels present in an image in a given direction ( $\Theta$ ) and distance ( $d$ ). GLCM is one of the original methods for extracting texture features; it was introduced by Haralick et al. (1973). The prior information of the images is obtained using GLCM. This texture analysis technique has several applications, one of which is the examination of medical images (Galloway, 1975). The  $(i, j)$ -th entry of  $G$  in GLCM, a square matrix  $G$  of order  $N$ , indicates how frequently an intensity  $i$  pixel is adjacent to an intensity  $j$  pixel. The normalized co-occurrence matrix is obtained by dividing each element of  $G$  by the total number of co-occurrence pairs in  $G$  (Chu et al., 1990). Adjacency is defined as the occurrence of a value in the horizontal, vertical, left, and right diagonal directions, as shown in the following Fig. 5. The matrix can be made symmetrical by adding it to its transpose and normalizing matrix  $G$  so that each cell represents the probability that that pair of intensities will occur in the image. Once the GLCM has been calculated, we can extract texture properties from the matrix to represent the textures in the image. The dimensions of our GLCM based feature vector for each image are  $4 \times 13 = 52$  since four directions ( $0^\circ, 45^\circ, 90^\circ, 135^\circ$ ) were taken into consideration for feature calculation, and for each direction, 13 texture features (as stated in Table 4) were computed.

$$P_{x+y}(k) = \sum_{\substack{i=1 \\ i+j=k}}^{N_g} \sum_{j=1}^{N_g} p(i, j), k=2, 3, 4, \dots, 2N_g \quad (7)$$

$$P_{x-y}(k) = \sum_{\substack{i=1 \\ |i-j|=k}}^{N_g} \sum_{j=1}^{N_g} p(i, j), k = 0, 1, 2, 3, 4, \dots, N_g - 1 \quad (8)$$

## 2.4 BPNN Model Used for Facial Image Classification

There are many classes in our database, and every class has forty face images. The size of the data grows to the point where applying the BPNN classifier requires a significant amount of memory. Therefore, the dataset was split up into several clusters, or groups, which are also known as super-classes, to facilitate the classification process for the BPNN Classifier. Data was clustered using a combination of hierarchical and K-Means clustering techniques.

Table 4. List of texture features calculated from GLCM matrix corresponding to CT image

S. No.	Texture Feature	Formula <sup>a</sup>	Description
1	Contrast	$\sum_{n=0}^{N_g-1} (i-j)^2 \left\{ \sum_{i=1}^{N_g} \sum_{j=1}^{N_g} p(i, j) \right\}$	Contrast is a measurement of the local variations in an image. It is the difference between the adjacent set of pixels' highest and lowest values based on frequency. A high contrast will be produced by a high degree of variation. 'Con' has a range of 0 to $(N_g-1)^2$ . An image that is constant has a value of 0.
2	Correlation	$\sum_{i=1}^{N_g} \sum_{j=1}^{N_g} \frac{ijp(i, j) - \mu_x \mu_y}{\sigma_x \sigma_y}$	The correlation between a pixel and its neighbor throughout the entire image is measured and returned by it. It has a range of -1 to +1. Correlation for an image is either 1 or -1 when it is perfectly positively or negatively correlated. Correlation for a constant image is NaN.
3	Uniformity (Angular Second Moment (ASM))	$\sum_{i=1}^{N_g} \sum_{j=1}^{N_g} p(i, j)^2$	The image's uniform distribution of grey levels is represented by ASM. ASM ranges from $1/N_g^2$ to 1. A value of 1 indicates a constant image. It establishes the degree of uniformity (or orderliness) in the gray level distribution of the image; images with a lower number of gray levels are more uniform.
4	Entropy	$\sum_{i=1}^{N_g} \sum_{j=1}^{N_g} p(i, j) \log(p(i, j))$	A statistical measure of randomness is called entropy. It falls between 0 and infinity. measures how disorganized the pixels in the image are. It is (roughly) inversely correlated with uniformity.
5	Homogeneity	$\sum_{i=1}^{N_g} \sum_{j=1}^{N_g} \frac{p(i, j)}{1 + (i - j)^2}$	A metric for the image's similarity is provided by homogeneity. The range of homogeneity is 0 to $N_g - 1$ . A visual similarity score of 0 indicates a high degree of similarity. It measures how homogeneous (smooth) the gray level distribution of the image is; it has an approximately inverse relationship with contrast, meaning that homogeneity is usually high when contrast is low.
6	Sum of squares (Variance)	$\sum_{i=1}^{N_g} \sum_{j=1}^{N_g} (i - \mu)^2 p(i, j)$	Variance quantifies the dispersion of the gray level distribution (relative to the mean).
7	Sum average <sup>b</sup>	$\sum_{i=2}^{2N_g} i p_{x+y}(i)$	It determines the gray level sum distribution mean of the image.

continued on following page

Table 4. Continued

S. No.	Texture Feature	Formula <sup>a</sup>	Description
8	Sum Variance <sup>b</sup>	$\sum_{i=2}^{2N_g} \left( i - \left[ \sum_{i=2}^{2N_g} i p_{x+y}(i) \right] \right)^2$	It computes the dispersion of the gray level sum distribution of the image (with respect to the mean).
9	Sum Entropy	$- \sum_{i=2}^{2N_g} p_{x+y(i)} \log \left\{ p_{x+y(i)} \right\}$	It quantifies the disorder associated with the image's gray level sum distribution.
10	Difference Variance <sup>c</sup>	$\sum_{i=2}^{2N_g} \left( i - \left[ \sum_{i=2}^{2N_g} i p_{x-y}(i) \right] \right)^2$	It measures the disorder related to the gray level difference distribution of the image.
11	Difference Entropy <sup>c</sup>	$\sum_{i=0}^{N_g-1} p_{x-y(i)} \log \left\{ p_{x-y(i)} \right\}$	It measures the disorder related to the distribution of gray level differences in the image.
12	Information measure of correlation-1	$F_{info.corr.1} = \frac{HXY - HXY_l}{HX}$ <p>Where,</p> $HXY = - \sum_{i=1}^{N_g} \sum_{j=1}^{N_g} p_{ij} \log_2 p_{ij}$ $HXY1 = - \sum_{i=1}^{N_g} \sum_{j=1}^{N_g} p_{ij} \log_2 (p_i \cdot p_j)$ $HX = - \sum_{i=1}^{N_g} p_i \cdot \log_2 p_i$	It measures the disorder related to the distribution of gray level differences in the image. It is employed to determine the background of the first measure of information correlation of the image applied to GLCM's gray level co-occurrence matrix. Two distinct metrics are used to estimate information theoretic correlation. The entropy for the joint probability is denoted by HXY. The entropy of the row marginal probability, denoted as HX, is symmetric with respect to the entropy of the column marginal probability. The entropy type HXY1 is calculated using the following formula:
13	Informaiton measure of correlation-2	$F_{info.corr.2} = \frac{\sqrt{1 - \exp(-2(HXY_2 - HXY))}}{HXY}$ $HXY = - \sum_{i=1}^{N_g} \sum_{j=1}^{N_g} p_{ij} \log_2 p_{ij}$ $HXY2 = - \sum_{i=1}^{N_g} \sum_{j=1}^{N_g} p_i \cdot p_j \log_2 (p_i \cdot p_j)$	When applied to GLCM, it calculates the image's second information-theoretic correlation measure.

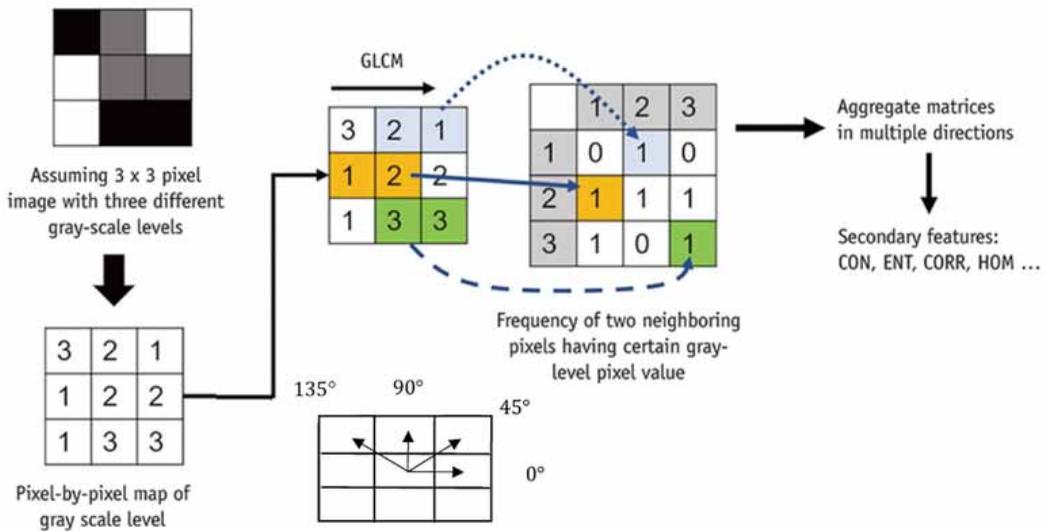
<sup>a</sup>In all above formula,  $p(i, j)$  is the  $(i, j)$ -th entry of the computed normalized GLCM, that is,  $p(i, j) = P(i, j) / \sum_{ij} P(i, j)$ , where  $P(i, j)$  is the  $(i, j)$ -th

entry of the computed GLCM;  $N_g$  is the total number of gray levels values present in the image; and  $\mu_x$ ,  $\mu_y$  and  $\sigma_x$ ,  $\sigma_y$  represent the computed mean and standard deviations of the row and column sums of the GLCM, respectively.

<sup>b</sup>The gray level sum distribution is given by Eqn. 7. It is related to the distribution of the sum of co-occurring pixels in the image.

<sup>c</sup>The gray level difference distribution is given by Eqn. 8. It is related to the distribution of the difference between co-occurring pixels in the image.

Figure 5. Computation process of GLCM feature matrix along with four directions (0°, 45°, 90°, 135°) and their secondary features from a sample digital image (Park et al., 2020)



#### 2.4.1 Building Super-Classes Using Combined K-Means and Hierarchical Clustering Methods

Each of the 117 subjects or classes in the NS-Database has 40 facial images. The hierarchical clustering method was used to normalize each class data set, in this case 40 data in each class. It generated two distinct clusters. Larger size clusters were taken into consideration for further developing super classes, while lower sized clusters were eliminated due to noisy or redundant data. To further cluster the representative dataset into super classes, the center of each class's larger cluster was selected as the class representative. K-Means clustering was used to create these super classes. Six super classes were created from the dataset's 117 distinct classes (6-value was heuristically determined after multiple trials, taking into account appropriate data distribution within super-classes as well as an adequate quantity of data availability within each super-class). Table 5 shows the number of classes in each super-class.

#### 2.4.2 Employing BPNN Classifier at Individual Super Class Level

The feed forward back-propagation algorithm, also known as the back-propagation neural network (BPNN) classifier, was utilized for classification using ANNs. Using the number of super-classes information—which included the number of output nodes equal to the number of super-classes—a BPNN classifier was applied to all of the data. Following the winner node concept of decision making, the testing data were convolved with the vector consisting of weights and biases obtained from the highest level BPNN classifier. Lastly, classification efficiency for all data related to its membership with its own super-class was computed as shown below in the result and discussion section. Similar

Table 5. Distribution of classes into their super classes using clustering approach for both the databases

	NS Database						ORL Database		
Super Class No.	1	2	3	4	5	6	1	2	3
Number of classes/subjects	8	17	17	28	36	11	15	15	10

to this, each super-class data set was now given individual consideration for BPNN classification; for example, six distinct BPNN classifiers were used for each of the six super-classes in our own NS database and three for the three super-classes in ORL database. In this instance, the number of real classes present in each super-class level corresponds to the number of output nodes used by the BPNN classifier. Because the hierarchical model used in this work has a two-level class hierarchy, BPNN classifiers are applied at two levels of the hierarchy. While the second level classifier provides information about the actual ground level class, the first level classifier makes the decision regarding the super-class information. Additionally, Fig. 1 and the introduction section 1 discussion help to clarify this.

### 2.4.3 Classification Accuracy of Object Recognition System

Any system's classification accuracy is calculated by dividing the total number of correctly classified data across all classes by the total number of data in the entire dataset. It is calculable using Eqn. 9:

$$\text{Classification Accuracy} = N_A / N \quad (9)$$

where,  $N_A$  = Number of data correctly classified among all classes and  $N$  = Total number of data present in the whole classes

### 2.4.4 Applying Clustering on Decisions Obtained From TFMLBPNN-DTC Based Classification

TFMLBPNN-DTC based classification used BPNN classifiers for each node (super-class and sub-class) at every decision tree level. One input set of data was used to classify using this model. To obtain a better decision or classification, human cognition approach was also applied in addition to TFMLBPNN-DTC based classification. This procedure was used in response to a decision made using BPNN classification. Since the human brain uses a group-based decision-making process (Wallis & Bülthoff, 2001), this approach was also used in our face recognition paradigm in place of making a decision based only on single test result. Decisions obtained by BPNN classification showing maximum accuracy on test data set were clustered into three clusters (number of clusters was heuristically chosen) using hierarchical clustering in order to determine the general tendency of classification decisions. It was assumed that the center of the largest cluster would provide the correct class decision and that the remaining clusters might contain incorrect data (methodology is also known). Clustering decisions, as per Kumar & Lahiri (2013b). In this work, it was assumed that correct test data might contain some noisy or incorrect data once it was discovered that, in accordance with traditional practice, the core TFMLBPNN-DTC based classifier accuracy on single test input was more than 80%. Additionally, it showed that, in the event that  $N$  of these data points were chosen at random and entered into the classifier, more than 80% of the decision vectors would result in accurate conclusions and, as a result, would cluster together within the decision space. Similar to this, if the noises in the noisy data are similar in kind, one might anticipate the formation of one more decision cluster. Based on this assumption, two clusters were created from the decision data, with the larger cluster containing the correct decisions and the smaller cluster containing noisy decisions. Thus, from its multiple inputs, the final decision regarding the class of an object (face, in this case) was obtained by taking the center of the largest cluster as representative of the correct decision, as illustrated in Fig. 6. Because this method did not depend on the initial condition, hierarchical clustering was applied to the data in this work. However, for ORL data, only one decision cluster was taken into account for each object (i.e., per person), due to the extremely low number of face images per person (total 10 considering both training and test data).

For instance, the hybrid ILMFD+GLCM feature that was obtained from a specific face image used as the test input produced a single decision (i.e., decision vector). In a similar vein, a set of

Figure 6. Decision flow in this work where clustering of decisions obtained from multiple input images of individual person to get center of largest cluster as final decision

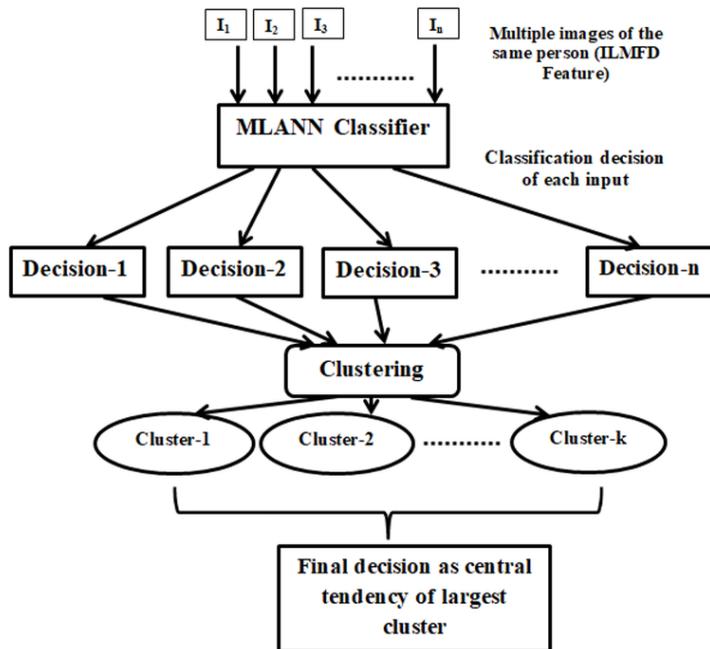


Table 6. Various performance metrics obtained from confusion matrix of our models

Confusion Matrix		Target			
		Positive	Negative		
Model	Positive	TP	FP	Positive Predictive Value (Precision)	TP/TP+FP
	Negative	FN	TN	Negative Predictive Value	TN/TN+FN
<b>F1-score=</b> $\frac{2 * \text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}}$		Sensitivity (Recall)	Specificity	<b>Accuracy =</b> $\frac{\text{TP} + \text{TN}}{\text{TP} + \text{FP} + \text{FN} + \text{TN}}$	
		$\frac{\text{TP}}{\text{TP} + \text{FN}}$	$\frac{\text{TN}}{\text{FP} + \text{TN}}$		

images, or test data, produced a set of several decisions, all of which had the same dimension since the classifier was left unchanged. The set of decisions was then carefully clustered again to produce the final decision, which came after the decision that matched the largest cluster’s central tendency. Numerous researchers also used the decision tree classification approach for a range of applications (Aisbett & Gibbon, 199; Karthigayani & Sridhar, 2014; Lafond et al., 2016; Luan et al., 2011).

### 3. EXPERIMENTAL RESULTS AND DISCUSSION

Tables 7 through 10 below describe the outcomes of the hybrid ILMFD+GLCM feature and TFMLBPNN-DTC based classifier used to classify the data in each super-class for our own NS database. Table 10 lists the classification accuracy for the combination of all super-classes (on the entire NS dataset). The classification results following the application of decision clustering (group test

Table 7. Parameters of BPNN architecture employed at each super class (on Our own NS database)

S. No.	Super class no.	Number of hidden layers	Number of inputs in input layer	Number of neurons in 1 <sup>st</sup> hidden layer	Number of neurons in 2 <sup>nd</sup> hidden layer	Number of neurons in output layer
1	1	1	62	35	10	8
2	2	1	62	35	10	17
3	3	1	62	35	10	17
4	4	1	62	35	10	28
5	5	1	62	35	10	36
6	6	1	62	35	10	11

Table 8. Classification results for each super class (on our own NS database) using BPNN and hybrid texture feature (ILMFD+GLCM)

Super class no.	Number of Classes out of total 117 classes	Train/test/validation data (70:15:15)	Training Algorithm	Total number of data (70:15:15)	Classification accuracy (%)	Precision (%)	Recall (%)	F1-score (%)	R <sup>2</sup>	Mean Square Error (MSE)
1	8	training	Levenberg–Marquardt algorithm (LMA)	224	100	100	100	100	1.00	0.0003
		test		48	100	100	100	100	0.95	0.0208
		validation		48	100	100	100	100	0.99	0.0050
2	17	training	Levenberg–Marquardt algorithm (LMA)	476	100	100	100	100	0.99	0.0008
		test		102	99.01	100	100	100	0.98	0.0069
		validation		102	100	100	100	100	0.99	0.008
3	17	training	Levenberg–Marquardt algorithm (LMA)	476	100	100	100	100	0.99	0.0018
		test		102	100	100	100	100	0.98	0.0046
		validation		102	100	100	100	100	0.98	0.0054
4	28	training	Levenberg–Marquardt algorithm (LMA)	784	100	100	100	100	0.93	0.0099
		test		168	99.40	100	100	100	0.89	0.0150
		validation		168	100	100	100	100	0.89	0.0154
5	36	training	Levenberg–Marquardt algorithm (LMA)	1008	100	100	100	100	0.99	0.0139
		test		216	100	100	100	100	0.89	0.0177
		validation		216	99.54	100	100	100	0.89	0.0173
6	11	training	Levenberg–Marquardt algorithm (LMA)	308	86.68	100	100	100	1.00	0.0074
		test		66	92.42	100	100	100	1.00	0.0027
		validation		66	74.24	100	100	100	1.00	0.0014
<b>Average</b>					<b>97.81</b>	<b>0.007432</b>	<b>100</b>	<b>100</b>	<b>0.96</b>	<b>0.0085</b>

data based decision) are also included in Table 10. Following the clustering of the classes into several super classes, the classification outcomes are displayed for each super-class independently. Tables 11 to 14 below describe the outcomes of the hybrid ILMFD+GLCM feature and BPNN classifier data classification at the super-class level for the ORL database. Table 14 lists the classification results for the variety of training and testing ratios for the combination of all super-classes (the entire ORL dataset). Additionally, it displays the classification outcomes following the application of decision

**Table 9. Classification results for super classes on our own NS database using BPNN and hybrid texture feature (ILMFD+GLCM) employed at root level of DCT**

Train/test/ validation data Ratio	Train/test/ validation data	Training Algorithm	Total number of data	Classification efficiency (%)	Precision (%)	Recall (%)	F1-score (%)	R <sup>2</sup>	Mean Square Error (MSE)
70:15:15	training	Levenberg–Marquardt algorithm (LMA)	3276	99.85	100	100	100	0.98	0.0105
	test		702	98.29	100	100	100	0.94	0.0239
	validation		702	98.15	100	100	100	0.94	0.0285
80:10:10	training	Levenberg–Marquardt algorithm (LMA)	3744	99.84	100	100	100	0.98	0.0092
	test		468	98.50	100	100	100	0.95	0.0255
	validation		468	99.15	100	100	100	0.95	0.0249
60:20:20	training	Levenberg–Marquardt algorithm (LMA)	2808	99.89	100	100	100	0.97	0.0091
	test		936	98.61	100	100	100	0.92	0.0332
	validation		936	99.03	100	100	100	0.94	0.0388
<b>Average</b>				<b>99.03</b>	<b>100</b>	<b>100</b>	<b>100</b>	<b>0.95</b>	<b>0.0226</b>

**Table 10. Classification results of proposed model using hybrid texture feature (GLCM+ILMFD) and MLPNN-DTC based classifier on our own NS database by considering different training/ testing ratio for both method i) using single test data ii) using cluster of test data (used for decision clustering tree approach)**

S No.	Training/Testing data ratio	Range of data in training(per subject)	Range of data in testing(per subject)	Cluster/Group/Individual testing data	Number of clusters	Total number of class representative data	Total number of class data correctly classified	Classification Accuracy	Precision (%)	Recall (%)	F1-score (%)	R <sup>2</sup>	Mean Square Error (MSE)
1	1:1	1:20	21:40	Cluster based	3	117	113	96.58	97.65	96.88	97.26	0.95	0.0258
2	1:1	1:20	21:40	Individual	--	2340	2218	94.79	95.45	94.90	95.17	0.92	0.0247
3	1:1	21:40	1:20	Cluster based	3	117	112	95.73	96.24	96.45	96.34	0.94	0.0198
4	1:1	21:40	1:20	Individual	--	2340	2216	94.70	94.90	95.24	95.07	0.93	0.0244
5	3:1	1:30	31:40	Cluster based	3	117	113	96.58	97.15	97.26	97.20	0.95	0.0135
6	3:1	1:30	31:40	Individual	--	1170	1123	95.98	96.05	95.80	95.92	0.94	0.0180
7	3:1	11:40	1:10	Cluster based	3	117	114	97.44	96.88	97.23	97.05	0.96	0.0120
8	3:1	11:40	1:10	Individual	--	1170	1132	96.75	96.80	97.35	97.07	0.95	0.0115
9	1:1	11:30	1:10 & 31:40	Cluster based	3	117	113	96.58	96.90	97.20	97.05	0.95	0.0224
10	1:1	11:30	1:10 & 31:40	Individual	--	2340	2214	94.62	95.50	94.80	95.15	0.93	0.0320
<b>Average metrics using single test data</b>								95.37	95.74	95.62	95.68	0.93	0.0221
<b>Average metrics using cluster based test data</b>								96.58	96.96	97.00	96.98	0.95	0.0187
<i>Average metrics ± Standard Deviation (using single test data)</i>								95.37 ±0.951877	95.74 ±0.718853	95.62 ±1.044088	95.68 ±0.851868	0.93 ±0.011401	0.0221 ±0.007733
<i>Average metrics ± Standard Deviation (using cluster based test data)</i>								96.58 ±0.604582	96.96 ±0.510029	97.00 ±0.345442	96.98 ±0.369527	0.95 ±0.007071	0.0187 ±0.005857

Table 11. Parameters of BPNN Architecture employed at each super class (on ORL database)

S. No.	Super class no.	Number of hidden layers	Number of inputs in input layer	Number of neurons in first hidden layer	Number of neurons in second hidden layer	Number of neurons in output layer
1	1	2	62	35	10	15
2	2	2	62	35	15	15
3	3	2	62	35	15	10

Table 12. Classification accuracy for each super class (ORL database)

S. No.	Super class no.	Train/test data	Total number of data	Total number of data correctly classified	Classification Accuracy (%)
1	1	training	75	74	98.67
		test	75	60	80.00
2	2	training	75	68	90.67
		test	75	57	76.00
3	3	training	50	49	98.00
		test	50	40	80.00

clustering (data-based group test). Following the clustering of all class/subject data into a number of super classes, the classification outcomes of each super-class’s unique BPNN classifiers are displayed separately. For the purpose of comparison, the entire project was tested on both NS and conventional ORL databases.

### 3.1 Performance Metrics Used to Measure the Performance of Proposed Models

Various metrics, also referred to as performance metrics or evaluation metrics, are employed to assess the model’s quality or performance. We can better understand our model’s performance for the provided data with the aid of these performance metrics. By adjusting the hyper-parameters, we can enhance the model’s functionality in this manner. Every machine learning model strives to achieve good generalization on new or unseen data, and performance metrics assess the model’s ability to do so.

**Confusion Matrix (CM):** A CM can be defined by a  $N \times N$  matrix, where  $N$  is the number of predicted classes. It is a metric used to assess how well machine learning classification problems perform when producing two or more classes as an output. A table with four distinct combinations of expected and actual values is called a confusion matrix as discussed below by Table 6, which are TP, FP, FN, and TN, Precision-recall, specificity, accuracy, and—most importantly—AUC-ROC curves can all be measured with great benefit from it.

The following metrics were used to evaluate the proposed models in this work:

**True Positive (TP):** our model predicted positive, and it’s true.

**True Negative (TN):** our model predicted negative, and it’s true.

**False Positive (FP) (Type 1 Error):** our model predicted positive, and it’s false.

**Table 13. Classification results of proposed model using hybrid texture feature (GLCM+ILMFD) and MLPNN-DTC based classifier on ORL face database by considering different training/ testing ratio for both method i) using single test data ii) by using cluster of test data (used for decision clustering tree approach)**

S.No.	Training/Testing data ratio	Range of data in training (per subject)	Range of data in testing(per subject)	Cluster/Group/Individual testing data	Number of clusters	Total number of class representative data	Total number of class data correctly classified	Classification Accuracy	Precision (%)	Recall (%)	F1-score (%)	R <sup>2</sup>	Mean Square Error (MSE)
1	7:1	1:7	7:10	Cluster based	1	40	36	92.5	91.62	92.45	92.03	0.90	0.0258
2	7:1	1:7	7:10	Individual	--	120	106	91.66	89.25	90.80	90.02	0.86	0.0427
3	7:1	4:10	1:3	Cluster based	1	40	34	87.5	86.44	86.35	86.39	0.84	0.0498
4	7:1	4:10	1:3	Individual	--	120	104	92.5	88.90	85.44	87.14	0.87	0.0364
5	1:1	1:5	6:10	Cluster based	1	40	36	92.5	89.15	91.46	90.29	0.88	0.0335
6	1:1	1:5	6:10	Individual	--	200	179	91	91.05	88.40	89.71	0.89	0.0280
7	1:1	6:10	1:5	Cluster based	1	40	37	95	91.88	93.13	92.50	0.90	0.0250
8	1:1	6:10	1:5	Individual	--	200	174	90	88.80	88.45	88.62	0.86	0.0315
9	1:1	4:8	1:3 & 9:10	Cluster based	1	40	35	90	87.90	89.40	88.64	0.85	0.0444
10	1:1	4:8	1:3 & 9:10	Individual	--	200	168	89	85.50	84.60	85.05	0.83	0.0441
<b>Average metrics using single test data</b>								90.83	88.70	87.54	88.11	0.862	0.0365
<b>Average metrics using cluster based test data</b>								91.50	89.40	90.56	89.97	0.874	0.0357
<b>Average metrics ± Standard Deviation (using single test data)</b>								<b>90.83</b> ±1.374369	<b>88.70</b> ±2.007175	<b>87.54</b> 2.512413	<b>88.11</b> ±2.047723	<b>0.862</b> ±0.021679	<b>0.0365</b> ±0.006954
<b>Average metrics ± Standard Deviation (using cluster based test data)</b>								<b>91.50</b> ±2.850439	<b>89.40</b> ±2.353342	<b>90.56</b> ±2.741199	<b>89.97</b> ±2.516555	<b>0.874</b> ±0.027928	<b>0.0357</b> ±0.011089

**Table 14. Comparison of performance metrics of our proposed models using both group based decision approach/individual test data on both NS and ORL database taking average of all variety of cases of train/test ratio**

Database	Cluster/Group/ Individual test data	Average Classification Accuracy (%) ± Standard Deviation	Average Precision (%) ± Standard Deviation	Average Recall (%) ± Standard Deviation	Average F1 Score (%) ± Standard Deviation	Average R <sup>2</sup> ± Standard Deviation	Average Mean Square Error ± Standard Deviation
NS	Individual	95.37 ±0.951877	95.74 ±0.718853	95.62 ±1.044088	95.68 ±0.851868	0.93 ±0.011401	0.0221 ±0.007733
	Group/Cluster based	96.58 ±0.604582	96.96 ±0.510029	97.00 ±0.345442	96.98 ±0.369527	0.95 ±0.007071	0.0187 ±0.005857
ORL	Individual	90.83 ±1.374369	88.70 ±2.007175	87.54 ±2.512413	88.11 ±2.047723	0.862 ±0.021679	0.0365 ±0.006954
	Group/Cluster based	91.50 ±2.850439	89.40 ±2.353342	90.56 ±2.741199	89.97 ±2.516555	0.874 ±0.027928	0.0357 ±0.011089

**False Negative (FN) (Type 2 Error):** our model predicted negative, and it's false.

**Accuracy:** the proportion of the total number of correct predictions by our model that were correct.

**Positive Predictive Value (Precision):** the proportion of positive cases of samples taken that were correctly identified by our model. When False Positives are more concerning than False Negatives, precision is helpful. Precision is crucial in e-commerce websites, music or video recommendation systems, and other systems where incorrect results could cause customer attrition and negatively impact the company's bottom line.

**Negative Predictive Value:** the proportion of negative cases of samples taken that were correctly identified by our model.

**Sensitivity (Recall):** the proportion of actual positive cases which are correctly identified by our model. When False Positive is more concerning than False Negative, recall is a valuable metric. In medical cases, it makes no difference if we sound a false alarm, but the real positive cases ought to be reported.

**Specificity:** the proportion of actual negative cases of samples taken which are correctly identified by our model.

**F score or F1 score:** It is a particular kind of single score that embodies both recall and precision. Therefore, the F1 Score can be computed by taking the harmonic mean of recall and precision and giving each equal weight. It should be used if both of them (precision as well as recall) are important for evaluation simultaneously.

**Mean Squared Error (MSE):** It calculates the mean of the squared difference between the values that the model predicts and the actual value that it provides. Since errors in MSE are squared, it can only assume values that are positive and typically non-zero. Additionally, because squared differences penalize small errors as well, the model's badness is overestimated. It is formulated by Eqn. 10:

$$MSE = \frac{1}{N} \sum (Y - \hat{Y})^2 \quad (10)$$

**R<sup>2</sup> Score:** It also called Coefficient of Determination (CoD), another widely used metric for evaluating machine learning models, is also known as R squared error. We can assess the model's performance by comparing it to a constant baseline using the R-squared metric. We must take the data mean and draw a line at the mean in order to choose the constant baseline. Whether the values are too big or too small, the R squared score will always be less than or equal to 1. It is formulated from the Eqn. 11.

$$R^2 = 1 - \frac{MSE(\text{Model})}{MSE(\text{Baseline})} \quad (11)$$

where, MSE is Mean Squared Error.

### 3.2 Classification at the Level of Super-Classes Using Hybrid ILMFD+GLCM Feature Criterion and BPNN Classifier on Non-Standard (NS) Database

Using a hybrid ILMFD+GLCM feature-based BPNN classifier, classification was carried out for each class independently. In addition, it was applied to the combination of all super classes, treating each super-class as a single class for classification purposes. The following is a discussion of the classification results for every case:

**Classification of super class-1:** There are thirty distinct subclasses in this superclass. There are forty face images of a specific person or subject in each subclass. Twenty images from each subclass were used for training, and the remaining twenty images were used for testing. Thus, in super class-1, a total of 20X30 data were used for testing and 20X30 for training.

Values of +1 and -1 were utilized for the target outputs. A value of +1 was assigned to each class to which the data belongs. If the data is not in that class, a value of -1 was used. A vector of weights and biases for the most well-known BPNN architecture was obtained after training the various BPNN classifier architectures through numerous trials. Three weight matrices and three bias matrices,  $b_1$ ,  $b_2$ , and  $b_3$ , were obtained as a result of the three layers. Since there are 30 actual classes/subjects in this super-class-1, there are 62 different inputs (a set of hybrid feature vector of size  $52+10=62$ , as discussed in the section-2.3.2.2 and 2.3.2.6) in the input layer, 30 neurons in the first hidden layer, 15 neurons in the second hidden layer, and 30 neurons in the final output layer. As a result, the sizes of the weights and biases are as follows:

$$w_1 = 30 \times 10, b_1 = 30 \times 1$$

$$w_2 = 15 \times 30, b_2 = 15 \times 1$$

$$w_3 = 30 \times 15, b_3 = 30 \times 1$$

where, for every connection between the input layer and the first hidden layer, is the weight matrix; likewise, for every connection between the first and second hidden layers, and for every connection between the second hidden layer and the final output layer, is the weight matrix. The vector called bias  $b_1$  contains the bias values for every node (neuron) in the first hidden layer,  $b_2$  represents the second hidden layer, and  $b_3$  represents the output layer. The super class-1 classification results for training, testing, and validation are shown in Table 8's first row.

**Classification of super class-2:** Following numerous trials, the well-known BPNN classifier model architecture was chosen for the super class-2 classification. The parameters in this architecture include the number of neurons in the input layer (10), first hidden layer (35), second hidden layer (20), and output layer (19). The classification results of super class-2 for training, testing, and validation are shown in Table 8's second row.

**Classification of super class-3:** The following well-known BPNN architecture was chosen for this super class, with ten input layers, thirty first hidden layers, fifteen second hidden layers, and nine output layers. The super class-3 classification results for training, testing, and validation are shown in Table 8's third row.

**Classification of super class-4:** The following well-known BPNN architecture was chosen for this super class, with 10 neurons in the input layer, 35 in the first hidden layer, 20 in the second hidden layer, and 18 in the output layer. The super class-4 classification results for training, testing, and validation are shown in Table 8's fourth row.

**Classification of super class-5:** The following well-known BPNN architecture, with ten input layers, forty first hidden layers, twenty-five second hidden layers, and twenty-six output layers, was chosen for this super class's classification. The super class-5 classification results for training, testing, and validation are shown in Table 8's fifth row.

**Classification of super class-6:** The following well-known BPNN architecture was chosen for this super class, with 10 neurons in the input layer, 35 in the first hidden layer, 20 in the second hidden layer, and 15 in the output layer. The super class-6 classification results for training, testing, and validation are shown in Table 8's sixth row.

### 3.2.1 TFMLBPNN-DTC Based Classification of Overall Actual Classes (117 Persons) on our own NS Database

The following well-known BPNN architecture, with the following numbers of neurons in each layer, was chosen for classification for this entire set of super-classes: Layers of input: 10, output: 6, and first and second hidden layers: 25 and 15, respectively. The classification results used at the decision clustering tree's root level to determine the super class number are displayed in Table 9.

Table 10 displays the group-based decision-making based on varying training and testing ratios as well as the classification results of the combination of all super classes on the NS Database for both scenarios. As indicated in Table 9, the overall accuracy ( $\pm$  standard deviation) of the suggested model on our own NS database is  $95.37 \pm 0.951877$  for individual input and  $96.58 \pm 0.604582$  for group-based decision. This finding demonstrates that using a cognition-based protocol has been shown to improve classification accuracy.

### 3.3 Classification at the Level of Super-Classes Using Hybrid ILMFD+GLCM Feature and BPNN Classifier on Standard ORL Database

Using a hybrid ILMFD+GLCM feature-based BPNN classifier, the same classification process was carried out for each class independently in the standard ORL database as well. Moreover, it was applied to a combination of all super classes, treating each super-class as a single class for classification purposes. The following is a discussion of the classification results for every case:

**Classification of super class-1:** There are fifteen distinct subclasses in this superclass. There are ten face images of a specific person or subject in each subclass. Five images from each subclass were used for training, and the remaining five images were used for testing. Thus, a total of  $5 \times 15 = 75$  data points were used in super class-1 for testing and  $5 \times 15 = 75$  for training.

Values of +1 and -1 were utilized for the target output. A value of +1 was assigned to each class to which the data belongs. If the data is not in that class, a value of -1 was used. A vector of weights and biases for the most well-known BPNN architecture was obtained after the BPNN was trained through numerous trials. We have three weight matrices and three bias vectors,  $b_1$ ,  $b_2$ , and  $b_3$ , because there are three layers. Since there are 15 actual classes in this super-class-1, there are 10 different inputs (a set of hybrid feature vector of size  $52 + 10 = 62$ , as discussed in sections 2.3.2.2 and 2.3.2.6) in the input layer, 35 neurons in the first hidden layer, 10 neurons in the second hidden layer, and 15 neurons in the final output layer. As a result, the sizes of the weights and biases are:

$$w_2 = 35 \times 10, b_1 = 35 \times 1$$

$$w_2 = 10 \times 35, b_2 = 10 \times 1$$

$$w_3 = 15 \times 10, b_3 = 15 \times 1$$

where all connections between the input layer and the first hidden layer are represented by the weight matrix  $w_1$ , all connections between the first and second hidden layers by  $w_2$ , and all connections between the second hidden layer and the final output layer by  $w_3$ . The vector called bias  $b_1$  contains the bias values for every node (neuron) in the first hidden layer,  $b_2$  represents the second hidden layer, and  $b_3$  represents the output layer. The classification results of super class-1 for ORL testing and training are shown in the first row of Table 12.

**Classification of super class-2:** The well-known BPNN classifier model's architecture was chosen for the classification after numerous trials. This architecture includes the following parameters:

input layer (10), first hidden layer (35), second hidden layer (15), and output layer (15). The super class-2 classification results for ORL testing and training are shown in Table 12's second row.

**Classification of super class-3:** The following well-known BPNN architecture was chosen for this super class, with 10 neurons in the input layer, 35 in the first hidden layer, 15 in the second hidden layer, and 10 in the output layer. The super class-3 classification results for ORL testing and training are shown in the third row of Table 12.

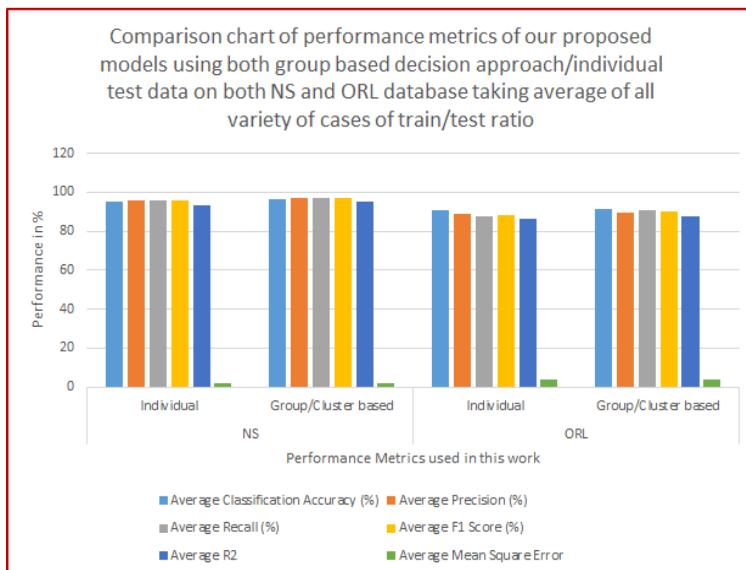
### 3.3.1 TFMLBPNN-DTC Based Classification of Overall Actual Classes (40 Persons) on ORL Database

The following well-known BPNN architecture, with the number of neurons in each layer as follows, was chosen for classification in the ORL database as well for the entire set of super-classes: Layers of input: 10, output: 3, first hidden layer: 25, and second hidden layer: 10. The classification results of all super classes combined on the ORL database are displayed in Table 13 for both group-based decisions based on varying training and testing ratios and scenarios involving a single input. As indicated in Table 13, the overall efficiency ( $\pm$  standard deviation) of the suggested model on the ORL database is  $90.83 \pm 1.374369$  for individual input and  $91.50 \pm 2.850439$  for group-based decision.

### 3.4 Summary of Classification Results Using MLPNN-DTC Classifier on Both our own NS Database and Standard ORL Database

Two clusters were used in this work to represent the number of outputs that corresponded to multiple inputs from the same person or class. The classification results for various training and testing ratios on the NS and ORL databases, respectively, were obtained by selecting the option that corresponded to the center of the largest cluster, as shown in Tables 8 and 13. The comparison of the suggested model's performance metrics is displayed in Table 14. The outcome suggests that a group/cluster-based approach to decision classification is more effective than a straightforward test approach with a single input. Fig. 7 depicts the comparison chart of performance metrics of our proposed models using both group based decision approach/individual test data on both NS and ORL database taking average of all metrics derived using variety of cases of train/test ratio.

Figure 7. Comparison chart of performance metrics of our proposed models using both group based decision approach/individual test data on both NS and ORL database taking average of all metrics derived using variety of cases of train/test ratio



Initially, it appears nearly impossible to successfully classify problems with a large number of classes. Conversely, humans have no trouble categorizing and distinguishing between thousands of objects, people, etc. The accuracy of cutting-edge computer vision and artificial intelligence techniques is getting closer to human accuracy, according to published research. Because of this principle of human cognition, in this work, multi-level BPNN classifiers inspired by cognition were used on all super classes that were derived from the dataset using hierarchical clustering and the similarity criteria of the hybrid feature extraction method. In order to improve the results, K-means clustering is taken into consideration when feeding the results from those BPNNs into decision clustering. This method relies on the largest cluster's center tendency to refer to the actual class decision among several decisions made in response to various input data points belonging to the same class. It also imitates human ability to recognize things quickly. With 40 facial photos per person and 117 people in the NS database, the suggested model is producing a classification accuracy  $\pm$  standard deviation of  $96.58 \pm 0.604582\%$ . Additionally, this model can be trained and tested for face recognition on a larger database of individuals. This method's application to other object recognition (such as sign language symbol classification) and classification problems is motivated by the notable improvement in classification results achieved for face detection and recognition. The fundamental concept of this work facilitates improved classification management, dealing with a wide range of classes.

### 3.5 Theoretical Analysis of the Computational Complexity of the Proposed MLPNN-DTC Model

Two face recognition models were created in this work: one for the non-standard database (NS), which contains 40 images of each of the 117 individuals in the database, and another for the ORL database (standard database), which contains 10 images of each individual in the database. Multi-level BPNNs were utilized in the development of the models at both the super-class and sub-class levels. Retraining the model is necessary if the number of participants is increased. In a similar vein, an object recognition system with 'N' objects can be created with that much data for training. After training, the model is prepared to perform a classification task and generate a result for an unknown input. Fig. 1 provides an understanding of this process. The classification task for any input in this model primarily consists of two parts:

- i) Feature extraction (hybrid ILMFD+GLCM) from input data: takes unit time  $\theta(1)$
- ii) Finding the actual class of input: similar to binary search  $\theta(\lg n)$

For example, let say the complete dataset is clustered into '3' super-classes (as shown in Fig. 1). In this case, there will be 4 classifiers employed in our model. Input-1 passes through classifier-1, which classifies its class as super-class-3 ( $\chi_3$ ). Now the classifier-4 which is employed on super-class-3 ( $\chi_3$ ), classifies input-1 to its actual class-3 (c3). Basically, whole classification task follows binary search process. Therefore, after trained model, the time complexity for classification task is:  $T(n) = \theta(1) + \theta(\lg n) = \theta(\lg n)$ . After the model is trained, the classification procedure only requires the storage of "n" vectors or matrices in terms of space complexity. The weights and biases of the final trained BPNN architectures are composed into vectors, or matrices, where "n" is the total number of BPNN classifiers used in the proposed model. The classification task of our proposed model with 'n' BPNN classifiers requires  $\theta(n)$  space if each vector takes up  $\theta(1)$  space.

## 4. CONCLUSION

The primary goal of the decision clustering approach is to address the successful mathematical translation of the human cognition principle proposed by Wallis and Bülthoff (2001) and Lycan (1999), who asserted that temporal descriptions of ill-posed objects improved recognition. A human

face was utilized as an example case study to improve the accuracy of their identification. In order to effectively apply the principle that advocates for the use of temporal, or video, data descriptions, a mathematical formulation of a multiple test-data based protocol was attempted in this work. This protocol operated by grouping intermediate decision vectors, each of which was derived from a single subject's test data. The Bayesian probabilistic model provides strong support for this strategy when it comes to repeating feature inputs. The noteworthy enhancement in efficiency achieved by our approach, which spans various features and databases, suggests that it warrants consideration for eventual implementation in suitable scenarios.

Additionally, the suggested model, which employs the decision tree clustering (DTC) approach, has demonstrated encouraging classification outcomes in the face recognition system. Our model produced  $95.37 \pm 0.951877\%$  and  $90.83 \pm 1.374369\%$  classification accuracy ( $\pm$ standard deviation) for single test input using MLPBPNN based decision tree clustering (TFMLBPNN-DTC) approach, and  $96.58 \pm 0.604582\%$  and  $91.50 \pm 2.850439\%$  for group based decision (decision clustering) for NS and ORL database, respectively. These outcomes demonstrate the potency of our suggested model, which combines multi-layer BPNN classifiers with hybrid cognition based texture features. This study also demonstrated the value of the DTC-based approach, which may be able to classify a sizable number of classes. This method divides the entire dataset into the desired number of clusters, or super-classes, based on factors such as easier processing and improved classification efficiency. The super-classes are then classified one at a time, and the final results corresponding to the entire data set are obtained by combining the classification outcomes of each super-class after it has been processed individually. This method, which was applied to the face recognition system in this work, can be applied to any type of data to improve classification. One of this model's advantages is that, after training, it needs to be saved in the system as vectors or matrices.  $\theta(n)$  computational space is needed for it. In this model, the classification task requires a time equal to binary search ( $\theta(\lg n)$ ) for any given input.

## 5. LIMITATIONS, SECURITY ISSUES, AND FUTURE SCOPE

In digital image processing, object detection and recognition—in this case, face recognition—are two interrelated tasks with a range of applications in social media, medical imaging, security, biometrics, entertainment, and satellite image analysis. Nevertheless, despite recent advancements in machine learning and machine vision, these tasks continue to face numerous obstacles and restrictions due to a range of factors, including face variability or diversity, the variety of object image databases that are available, algorithm complexity and trade-offs, privacy and ethics, and future directions. The great variability and diversity of human faces with regard to size, shape, pose, illumination, expression, occlusion, resolution, and makeup is a major drawback for any face recognition system. These factors vary, which makes it difficult for the algorithms in use to generalize, handle, and deal with a variety of scenarios and conditions. For instance, when faces in the images are partially obscured by masks, hands, or hats, or when the faces are twisted, translated, tilted, or otherwise distorted, the face detection system's performance suffers. For example, faces that are not aligned, segmented, cropped, pre-processed, or normalized, or faces that change over time as a result of aging, burning, facial hair, beards, or cosmetics, have all been linked to low performance in these systems. There are some security and ethical issues in facial recognition system such as improper data storage, misuse of data, violation on individual privacy, infringement on freedom of speech and association, lack of transparency, accessible to everyone etc. These concerns need to be handled properly in order to build a secured biometric system.

There are certain drawbacks and difficulties with our suggested model. However, the facial image databases (both NS and ORL) used in this work include some frames that were taken from face motion-based videos (180° rotation), as well as facial images captured at various times, locations, and lighting conditions. There is also variation in orientation, facial expressions (open vs closed eyes, smiling vs

not smiling), and external facial details. There are still some scenarios that can be added to create a more resilient system and increase the employing algorithms' adaptability to previously unexplored and emerging subjects, domains, and scenarios. Furthermore, a scale and rotation independent hybrid ILMFD+GLCM feature was employed for feature extraction. This feature consists of the fractal dimension of ten equal intensity interval based image slices, and it is assumed to capture the majority of the various intensity regions found in facial images. To extract more precise information from an image, more image slices may be used in subsequent work. Similarly for GLCM feature, 4 different directions were taken. Further number of directions can be increased. For the proposed model to perform better, more texture-based feature extraction techniques can be investigated, such as Binary Robust Independent Elementary Features (BRIF), First Order Statistics (FOS), Gray-Level Run-Length Matrix (GLRLM), Gray-Level Size Zone Matrix (GLSZM), Neighboring Gray Tone Difference Matrix (NGTDM), Scale-Invariant Feature Transform (SIFT), Gray Level Dependence Matrix (GLDM), Speeded-Up Robust Features (SURF), First Order Statistics (FOS), Binary Robust Independent Elementary Features (BRIF), etc. Another difficulty is that a large number of images must be collected for each person in order to properly train the BPNN model for object recognition. In a similar vein, additional machine learning or deep learning techniques may be investigated to improve face recognition system classification. Using the same methodology, automated systems for identifying different kinds of objects can be created in the future.

## **ACKNOWLEDGMENT**

The author would like to give thanks to the owners of the ORL facial database that was used in this work for comparison work. The author is also thankful to all the persons, who had given concern to allow to capture the facial images to create our own NS database for the study in this work.

## **CONFLICT OF INTEREST**

The authors declare that they have no conflict of interest

## **DATA AVAILABILITY STATEMENT**

The datasets generated during and/or analyzed during the current study are available from the corresponding author on reasonable request.

## REFERENCES

- Abdullahi, S. M., Wang, H., & Li, T. (2020). Fractal Coding-Based Robust and Alignment-Free Fingerprint Image Hashing. *IEEE Transactions on Information Forensics and Security*, 15, 2587–2601. doi:10.1109/TIFS.2020.2971142
- Aisbett, J., & Gibbon, G. (1999). *Cognitive Classification*. Proceedings of the sixteenth national conference on Artificial intelligence and the eleventh Innovative applications of artificial intelligence conference innovative applications of artificial intelligence, Orlando, Florida, USA.
- Alazawi, S. A., Shati, N. M., & Abbas, A. H. (2019). Texture features extraction based on GLCM for face retrieval system. [PEN]. *Periodicals of Engineering and Natural Sciences*, 7(3), 1459. doi:10.21533/pen.v7i3.787
- Alsmadi, M., Omar, K., & Noah, S. (2009). Back Propagation Algorithm: The Best Algorithm among the Multi-layer Perceptron Algorithm. *IJCSNS International Journal of Computer Science and Network Security*, 9(4), 378–383.
- Atkinson, P., & Tatnall, A. (1997). Introduction Neural Networks in Remote Sensing. *International Journal of Remote Sensing*, 18(4), 699–709. doi:10.1080/014311697218700
- Bisogni, C., Nappi, M., Pero, C., & Ricciardi, S. (2020). *HP2IFS: Head Pose estimation exploiting Partitioned Iterated Function Systems*. 2020 25th International Conference on Pattern Recognition (ICPR), Milan, Italy. doi:10.1109/ICPR48806.2021.9413227
- Bruce, V., & Young, A. (1998). *In the eye of the beholder. The science of face perception*. Oxford University Press Inc.
- Chaudhuri, B. B., & Sarkar, N. (1995). Texture segmentation using fractal dimension. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(1), 72–77. doi:10.1109/34.368149
- Cheng, S. C., & Yueh-Min, H. (2003). A novel approach to diagnose diabetes based on the Fractal characteristics of retinal images. *IEEE Transactions on Information Technology in Biomedicine*, 7(3), 163–170. doi:10.1109/TITB.2003.813792 PMID:14518729
- Chu, A., Sehgal, C. M., & Greenleaf, J. F. (1990). Use of gray value distribution of run lengths for texture analysis. *Pattern Recognition Letters*, 11(6), 415–419. doi:10.1016/0167-8655(90)90112-F
- Deaton, R., Tang, L., & Reddick, W. E. (2002). Fractal analysis of magnetic resonance images of the brain. *Proceedings of 16th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*. IEEE. doi:10.1109/IEMBS.1994.411875
- Duolin, L., & Wei, T. (2022). Deep learning-based image target detection and recognition of fractal feature fusion for BIOmetric authentication and monitoring. *Network Modeling and Analysis in Health Informatics and Bioinformatics*, 11(1), 17. doi:10.1007/s13721-022-00355-5
- Ebrahimpour, R., Ehteram, S. R., & Kabir, E. (2005). Face Recognition by Multiple Classifiers, a Divide-and-Conquer Approach, Knowledge-Based Intelligent Information and Engineering Systems. Lecture Notes in Computer Science. Springer.
- Ebrahimpour-Komleh, H., Chandran, V., & Sridharan, S. (2001). Robustness to expression variations in Fractal-based Face recognition, Sixth International. *Symposium on Signal Processing and its Applications*, Kuala Lumpur.
- Feder, J. (1989). *Fractals*. Plenum Press.
- Ferens, K., & Kinsner, W. (1995). Multifractal texture classification of images. *IEEE Proceedings, Conference on Communication, Power, and Computing*, (Vol. 2, pp. 438 – 444). IEEE.
- Galloway, M. M. (1975). Texture analysis using gray level run lengths. *Computer Graphics and Image Processing*, 4(2), 172–179. doi:10.1016/S0146-664X(75)80008-6
- Ghaderi S (2023). Fractal Dimension Image Processing for Feature Extraction and Morphological Analysis: Gd3+/13X/DOX/FA MRI Nanocomposite. *Journal of Nanomaterials*. .10.1155/2023/8564161
- Haralick, R. M., Shanmugam, K., & Dinstein, I. (1973). Textural features for image classification. *IEEE Transactions on Systems, Man, and Cybernetics, SMC–3*(6), 610–621. doi:10.1109/TSMC.1973.4309314

- Hossein, E. K. V., Chandran, V., & Sridharan, S. (2001). Face recognition using fractal codes, Conference: Image Processing Proceedings. *2001 International Conference*. IEEE. doi:10.1109/ICIP.2001.958050
- Jiang, J., Zhang, J., Yang, G., Zhang, D., & Zhang, L. (2010). Application of Back Propagation Neural Network in the Classification of High Resolution Remote Sensing Image: Take Remote Sensing Image of Beijing for Instance. In *Proceedings of 18th International Conference on Geoinformatics*. IEEE. doi:10.1109/GEONFORMATICS.2010.5568228
- Kaewchinporn, C. (2011). A combination of decision tree learning and clustering for Data Classification. *Eighth International Joint Conference on Computer Science and Software Engineering (JCSSE)*, Nakhon Pathom. doi:10.1109/JCSSE.2011.5930148
- Karthigayani, P., & Sridhar, S. (2014). Decision tree based occlusion detection in face recognition and estimation of human age using back propagation neural network. *Journal of Computational Science*, 10(1), 115–127. doi:10.3844/jcssp.2014.115.127
- Karthigayani, P., & Sridhar, S. (2014). Decision tree based occlusion detection in face recognition and estimation of human age using back propagation neural network. *Journal of Computational Science*, 10(1), 115–127. doi:10.3844/jcssp.2014.115.127
- Keller, J. M., Chen, S., & Crownover, R. M. (1989). Texture description and segmentation through fractal geometry. *Computer Vision Graphics and Image Processing*, 45(2), 150–166. doi:10.1016/0734-189X(89)90130-8
- Khan, M. N. A., Qureshi, S. A., & Riaz, N. (2013). Gender Classification with Decision Trees, International Journal of Signal Processing. *Image Processing and Pattern Recognition*, 6(1), 165–176. doi:10.1109/CGiV.2016.33
- Knight, B., & Johnston, A. (1997). The role of movement in Face recognition. *Visual Cognition*, 4(3), 265–274. doi:10.1080/713756764
- Kouzani, A. Z., He, F., & Sammut, K. (1997). Fractal faces representation and recognition. *IEEE International Conference on Systems, Man, and Cybernetics*. IEEE.
- Kumar, U., & Lahiri, T. (2013a). Segmentation of ill-defined objects by convoluting context window of each pixel with a non-parametric function. *International Journal of Computer Vision and Image Processing*, 3(1), 33–41. doi:10.4018/ijcvip.2013010103
- Kumar, U., & Lahiri, T. (2013b). Significant Enhancement of Object Recognition Efficiency Using Human Cognition based Decision Clustering. *International Journal of Computer Vision and Image Processing*, 3(4), 1–15. doi:10.4018/ijcvip.2013100101
- Lafond, D., Vallières, B. R., Vachon, F., St-Louis, M.-E., & Tremblay, S. (2016). Capturing Non-linear Judgment Policies Using Decision Tree Models of Classification Behavior. *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, 59(1), 831–835. doi:10.1177/1541931215591251
- Li, H. Q., Chen, S. H., & Zhao, H. M. (1991). Fat fractal and multifractals for protein and enzyme surfaces. *International Journal of Biological Macromolecules*, 13(4), 210–216. doi:10.1016/0141-8130(91)90074-5 PMID:1777427
- Luan, S., Schooler, L. J., & Gigerenzer, G. (2011). A signal detection analysis of fast-and-frugal trees. *Psychological Review*, 118(2), 316–338. doi:10.1037/a0022684 PMID:21381858
- Lycan, W. G. (1999). *Mind and Cognition: An Anthology* (2nd ed.). Blackwell Publishers, Inc.
- Mandelbrot. (1983). *The fractal geometry of nature*.
- Muqasqas, S. A., Radaideh, Q. A. A., & Abul-Huda, B. A. (2014). A Hybrid Classification Approach Based on Decision Tree and Naïve Bays Methods. *International Journal of Information Retrieval Research*, 4(4), 61–72. doi:10.4018/IJIRR.2014100104
- Park, H. J., Park, B., & Lee, S. S. (2020). Radiomics and deep learning: Hepatic applications. *Korean Journal of Radiology*, 21(4), 387–401. doi:10.3348/kjr.2019.0752 PMID:32193887

- Pepik, B., Kalajdziski, S., & Davec, D. (2009). Protein Classification Using Decision Trees With Bottom-up Classification Approach. *13th International Conference on Biomedical Engineering*. Springer. doi:10.1007/978-3-540-92841-6\_42
- Ranganath, A., & Mishra, J. (2017). *New approach for estimating fractal dimension of both gray and color images*. In: *IEEE 7th International Advance Computing Conference (IACC)*, Hyderabad. doi:10.1109/IACC.2017.0142
- Ranganath, A., Senapati, M. R., & Sahu, P. K. (2021). Estimating the fractal dimension of images using pixel range calculation technique. *The Visual Computer*, 37(3), 635–650. doi:10.1007/s00371-020-01829-1
- Ren, N., Zargham, M., & Rahimi, S. (2006). A Decision Tree-Based Classification Approach to Rule Extraction for Security Analysis. *International Journal of Information Technology & Decision Making*, 5(1), 227–240. doi:10.1142/S0219622006001824
- Salmam, F. Z., Madani, A., & Kissi, M. (2016). Facial Expression Recognition using Decision Trees. *13th International Conference on Computer Graphics, Imaging and Visualization (CGiV)*, (pp. 125-130). IGI. doi:10.1109/CGiV.2016.33
- Salman, A. D., Talab, M. A., & Al-Dahhan, R. R. (2022). Features extraction for robust face recognition using GLCM and CS-LBP. In *Lecture Notes in Networks and Systems. Proceedings of International Conference on Emerging Technologies and Intelligent Systems* (pp. 175–191). . <https://doi.org/> doi:10.1007/978-3-030-85990-9\_16
- Schmidhuber J (1998) *Facial beauty and Fractal geometry*. Note IDSI A-28-98.
- Shanmugavadivu, P., & Sivakumar, V. (2012). Fractal Dimension Based Texture Analysis of Digital Images. *Procedia Engineering*, 38, 2981-2986. 10.1016/j.proeng.2012.06.348
- Singh, R., Samal, S., & Lahiri, T. (2005). A Novel Strategy for Designing Efficient Multiple Classifier. *Lecture Notes in Computer Science*, 3832, 713–720. doi:10.1007/11608288\_95
- Tan, T., & Yan, H. (1999) *Face recognition by Fractal transformation*. *Proceedings, 1999 IEEE International Conference on Acoustics, Speech, and Signal Processing*, Phoenix, AZ.
- Tripathi, E., Kumar, U., & Tripathi, S. P. (2022). Image splicing detection system using intensity-level multifractal dimension feature engineering and twin support vector machine based classifier. *Multimedia Tools and Applications*.10.1007/s11042-022-13519-2
- Tsai, C. A., Chen, D. T., Chen, J. J., Balch, C. M., Thompson, J. F., & Soong, S. J. (2007). An integrated tree-based classification approach to prognostic grouping with application to localized melanoma patients. *Journal of Biopharmaceutical Statistics*, 17(3), 445–460. doi:10.1080/10543400701199585 PMID:17479393
- Vera, K., A., Pane, I. Z., Overbeek, M. V., & Prasetya, S. G. (2023, February 5). *Face recognition accuracy improving using gray level co-occurrence matrix selection feature algorithm*. Presented at the 2023 International Conference on Smart Computing and Application (ICSCA), Hail, Saudi Arabia. doi:10.1109/ICSCA57840.2023.10087414
- Wallis, G. M., & Bühlhoff, H. H. (2001). Effects of temporal association on recognition memory. *Proceedings of the National Academy of Sciences of the United States of America*, 98(8), 4800–4804. doi:10.1073/pnas.071028598 PMID:11287633
- Wilson, C. L., Grother, P. J., & Barnes, C. S. (1996). Binary decision clustering for neural-network based optical character recognition. *Pattern Recognition*, 29(3), 425–437. doi:10.1016/0031-3203(95)00105-0
- Xie, H. A., Wang, J. A., & Stein, E. (1998). Direct fractal measurement and multifractal properties of fracture surfaces. *Physics Letters. [Part A]*, 242(1-2), 41–50. doi:10.1016/S0375-9601(98)00098-X

*Upendra Kumar is working as Assistant Professor in Department of Computer Science and Engineering, in Institute of Engineering and Technology, Lucknow, India (Renowned UP Government Engineering Institute established since 1984). He has done M.Tech. in IT from IIT Allahabad, and Ph.D in Computer Science & Engineering from Dr. A.P.J. Abdul Kalam Technical University, UP, Lucknow. He has excellent academic record from 10th standard to Master program throughout first honours. He has around four years industrial experience in renowned organization in TCS and worked there in many corporate projects as a JAVA/J2EE developer. He has more than 15 years teaching and research experience. His area of research interest is Digital Image Processing, Machine Learning, Biomedical Signal Processing, Fractal and Chaos Analysis, Robotics and Embedded in C, Medical Imaging and Diagnostic System, Big Data Analytics etc. He has published 30+ international and national journal and conference research papers with quality research outcomes. He has published one Indian patent and one germany patent. He has also published 02 edited book in Taylor & Francis and 15+ book chapters in international level journals like IGI Global publisher, Taylor & Francis and etc. Currently four research scholars are doing research under his supervision and one student has been awarded his Ph.D degree. He has delivered 20+ expert lectures in various international and national conferences, workshops, seminars and short term courses. He has also created video lectures on various computer science subjects, uploaded in YouTube and got very good responses from the viewers. He is member of various review committees of national and international journals, conferences, professional bodies. Apart from teaching and research work, he has also some administrative role working as Deputy Dean of Students Welfare in the Institute (IET Lucknow) and Associate Dean of Computer Application at AKTU Lucknow.*