IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received 17 August 2023, accepted 26 September 2023, date of publication 5 October 2023, date of current version 21 March 2024.

Digital Object Identifier 10.1109/ACCESS.2023.3322147

APPLIED RESEARCH

Adaptive Feature Selection and Image Classification Using Manifold Learning Techniques

AMNA ASHRAF[1,2], NAZRI MOHD NAWI[2], AND MUHAMMAD AAMIR[3]

[1]Department of Artificial Intelligence, The Islamia University of Bahawalpur, Bahawalpur 63100, Pakistan
[2]Faculty of Computer Science and Information Technology, Universiti Tun Hussein Onn Malaysia (UTHM), Parit Raja, Johor 86400, Malaysia
[3]School of Electronics, Computing and Mathematics, University of Derby, DE22 3AW Derby, U.K.

Corresponding authors: Amna Ashraf (amna.ashraf@iub.edu.pk) and Nazri Mohd Nawi (nazri@uthm.edu.my)

ABSTRACT Manifold learning techniques aim to the non-linear dimension reduction of data. Dimension reduction is the field of interest and demand of many data analysts and is widely used in computer vision, image processing, pattern recognition, neural networks, and machine learning. The research has been divided into two phases to recognize manifold learning techniques' importance. In the first phase, the manifold learning approach is used to improve the 'feature selection by clustering'. Clustering algorithms such as K-means, spectral clustering, and the Gaussian Mixer Model have been tested with manifold learning approaches for adaptive feature selection. The results obtained are satisfactory compared to simple clustering. In the second phase, a Triple Layered Convolutional Architecture (TLCA) has been proposed for image classification bearing 85.34%, 59.14%, 71.43%, 90.06%, and 71.71% accuracy levels for the datasets such as Pistachio, Animal, HAR, Mango Leaves, and Cards respectively. The performance of the proposed TLCA model is compared to the other deep learning models i.e., CNN, LSTM, and GRU. To further improve the accuracy, reduced dimensional data from manifold learning technique is used and achieved higher accuracies from Hybrid Triple Layered Convolutional Architecture HTLCA as 97.73%, 87.18%, 97.97%, 99.19%, and 96.91% for the mentioned sequence of datasets. The effectiveness and precision of the suggested methods are demonstrated by the experimental findings.

INDEX TERMS Clustering, feature extraction, feature selection triple layered convolutional architecture.

I. INTRODUCTION

Manifold learning is a machine learning and data analysis technique that extracts meaningful features from high-dimensional data [1]. Its primary objective is to identify a lower-dimensional representation of the data that preserves the underlying structure and relationships among the data points. The technique treats the data as it lays on a manifold, which is a curved, lower-dimensional surface embedded in the high-dimensional space. This manifold can be envisioned as a twisted or folded version of the high-dimensional space. By identifying the underlying manifold, the manifold learning algorithms can uncover the intrinsic structure of the data, thereby extracting meaningful features that capture this structure.

Several techniques for manifold learning exist, including Principal Component Analysis (PCA) [2], t-distributed Stochastic Neighbor Embedding (t-SNE) [3], and Isometric Mapping (Isomap) [4], among others. These techniques employ different algorithms to determine the lower-dimensional representation of the data while preserving the relationships among the data points. The extracted features can be utilized for diverse tasks, such as classification, clustering, object recognition, image retrieval, and visualization [5]. Through the reduction of the data's dimensionality and the extraction of meaningful features, manifold learning enhances the performance of machine learning and deep learning algorithms and simplifies the understanding and interpretation of the data.

The associate editor coordinating the review of this manuscript and approving it for publication was Jeon Gwanggil.

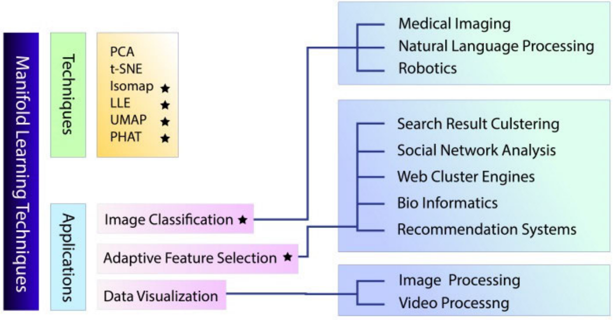


FIGURE 1. Main focus of the research.

The use of extracted features from manifold learning techniques helps deep learning algorithms to accurately classify images based on their underlying structure and relationships, thereby improving the performance of computer vision systems [6]. Image classification by deep learning algorithms is widely used in fields such as medical imaging, natural language processing, and robotics. By reducing the dimensionality of the image data and extracting meaningful features, manifold learning can enhance the performance of computer vision systems, thereby advancing research and practical applications in these fields.

Diffusion maps [7], Laplacian eigenmaps, and manifold regularized extreme learning machines [8] are other manifold learning algorithms that have been used for picture categorization. These techniques have shown a promising increase in image classification accuracy as they aim to capture many features of the underlying data structure. A researcher proposed a unique approach to feature selection that makes use of both labeled and unlabeled data [9]. To find the most pertinent features for classification, a strategy that combines manifold learning with a graph-based semi-supervised learning algorithm is used. To propagate labels from labeled to unlabeled data, it uses the graph-based semi-supervised learning algorithm.

Feature selection is a data preprocessing technique that prepares data for various data mining and machine learning tasks [10]. It aims a simpler and more comprehensive model to improve data mining performance and produce clean and logical data. In recent decades, numerous feature selection techniques have been introduced, primarily designed for supervised classification problems. However, the recent advancements in technology and the abundance of unlabeled data generated in various applications, such as text mining, image retrieval, social media and intrusion detection, have led to a significant interest in Unlabeled Feature Selection (UFS) methods within the scientific community. Daniela proposed a solution for feature selection SFAM [11] a unified learning paradigm that combines adaptive global structure learning with manifold learning, to address the algorithm cost concern. The method is designed to retain global and sparse reconstruction structure while investigating local structure and label correlations.

The main attention of this study is to realize the importance of manifold learning techniques in the domain of machine learning and deep learning. The focus of our research has been mentioned in Fig. 1 and the techniques used and applications considered are stared in the diagram. The major applications are image classification, adaptive feature selection, and data visualization. Image classification is used in the fields of medical imaging, natural language processing, and robotics. The adaptive feature selection technique has its advantages in web cluster engines [12], bioinformatics [13], recommendation systems, search result clustering, and social network analyses, while data visualization is essential for image and video processing [14].

There are lots of feature selection methods that already exist like filters, wrappers, and some hybrid methods [15]. Clustering itself facilitates feature selection. Different clustering algorithms have different accuracies on different datasets. These accuracies can be improved using manifold learning techniques. The same is the case with image classification. Experimental results show that introducing feature extraction by manifold learning can play an important role in adaptive feature selection and perform better image classification than that can be achieved by state-of-the-art deep learning models.

II. PRELIMINARIES

Machine learning and data analysis employ manifold learning approaches to comprehend and extract high-dimensional data structures. Deep learning techniques have excelled by outperforming other techniques in a variety of applications, including text mining, speaker identification, handwriting recognition and object detection and recognition. Data often resides on a lower-dimensional manifold embedded in a higher-dimensional environment in real-world applications. Manifold learning attempts to capture and explain this fundamental structure. Different manifold learning techniques are discussed and elaborated below.

A. ISOMAP

Isomap dimensionality reduction preserves geodesic distances between data points. Visualizing high-dimensional data in smaller dimensions is typical. Isomap creates a neighborhood graph from paired data point distances and finds a low-dimensional embedding that retains geodesic distances. Isomap has the following steps to be followed. Data input X with 'd' dimensions refers to (1) having 'n' number of data points.

$$X = [x_1, x_2, \ldots, x_n], x_i : R^d \tag{1}$$

Pairwise distances between data points are computed to build the neighborhood graph. The distance matrix $D = [d_{ij}]$ shows the distance between data points x_i and x_j. In the k-nearest neighborhood graph G, Euclidean distance (2) is used to calculate the edge length

$$\mathrm{d_{ij}} = \left\|\mathrm{x_i} - \mathrm{x_j}\right\|^2 \tag{2}$$

An adjacency matrix A represents the neighborhood graph, where Aij = 1 if $\mathrm{x_i}$ and $\mathrm{x_j}$ are connected and 0 otherwise. Next, the geodesic distances between all pairs of data points are obtained. The shortest route distance along G is the geodesic distance (3) $\mathrm{G_{ij}}$ between $\mathrm{x_i}$ and $\mathrm{x_j}$.

$$G_{ij} = shortest\_path\_distance(x_i, x_j) \tag{3}$$

The shortest paths are usually calculated using graph-based methods like Dijkstra's or Floyd-Warshall. The distance matrix $\mathrm{D^G} = [\mathrm{G^{ij}}]$ reflects the geodesic distance between $\mathrm{x_i}$ and $\mathrm{x_j}$. Isomap computes a low-dimensional data embedding using classical multidimensional scaling (MDS). MDS finds a group of points in a lower-dimensional space that approximates pairwise distances from the high-dimensional space. The low-dimensional embedding matrix $\mathrm{Y} = [\mathrm{y_1}, \mathrm{y_2}, \ldots, \mathrm{y_n}]$ represents the lower-dimensional coordinates of each data point $\mathrm{x_i}$.

Isomap has proved successful in several applications. Data and parameters determine its efficacy. Isomap, like other dimensionality reduction methods, does not function well for all datasets. The data structure, noise, and outliers affect its performance.

B. LLE

Locally Linear Embedding (LLE) is an effective non-linear dimension reduction technique for reducing the features of high-dimensional data while retaining its core geometric structure. The LLE algorithm consists of three key stages: constructing a neighborhood graph, computing the weight matrix, and computing the embedding coordinates. To begin, the algorithm constructs a neighborhood graph G represented by an adjacency matrix. It identifies the k nearest neighbors 'j' of each data point 'i' and connects them with edges. The variable '$\mathrm{G_{ij}}$' sets to 1 if there is an edge between i and j otherwise sets to 0. Next, for each data point, the algorithm computes a weight matrix E(W) (4) by minimizing the reconstruction error between the data point and its neighbors using linear weights $\mathrm{W_{ij}}$.

$$\mathrm{E\,(W)} = \sum\nolimits_{\mathrm{i}} \left|\mathrm{X_i} - \sum\nolimits_{\mathrm{j}} \mathrm{W_{ij} X_j}\right|^2 \tag{4}$$

Finally, the algorithm computes the embedding coordinates Yi by minimizing a cost function C(Y) that preserves the local relationships between the data points referred to (5).

$$\mathrm{C\,(Y)} = \sum\nolimits_{\mathrm{i}} \left|\mathrm{Y_i} - \sum\nolimits_{\mathrm{j}} \mathrm{W_{ij} Y_j}\right|^2 \tag{5}$$

The resulting embedding coordinates provide a lower-dimensional representation of the data that maintains its essential geometric structure. LLE recovers global nonlinear structure from locally linear fits, unlike Isomap.

C. UMAP

UMAP is mostly used for larger datasets to convert high dimensional data to lower dimensional data that visualization is much better and easy. It is beneficial for the outliers and similarities to be identified. UMAP works in a way that preserves the high-dimensional grouping of data and the relationships between different data points. The method starts with all the high dimensional points in low dimension and then moves those low dimensional data points so that the categorization among different groups remains as same as the relationships present in high dimension data. Distances between every pair of data in high dimensions are calculated in the initial step. Then UMAP algorithm determines the similarity score for each cluster which helps recognize how good clustering has been done. It must be as same as the clusters in the low-dimensional graph present. UMAP uses Spectral Embedding to initiate a low-dimensional graph by using the similarity score SS (6).

$$SS = e^{-(Raw\ Distance - Distance\ to\ nearest\ neighbor)/\sigma} \tag{6}$$

$$\mathrm{Cost} = \log\left(\frac{1}{\mathrm{neighbor}}\right) + \log\left(\frac{1}{1 - \mathrm{notneighbor}}\right) \tag{7}$$

UMAP focuses on the two scores 'neighbor' and 'not neighbor' to evaluate if a point is in the right place or not. There is a Cost Function elaborated in (7) which uses the two scores to calculate. For an optimal low-dimensional graph very few points are moved at a time by Stochastic Gradient Descent.

D. PHATE

High-dimensional data is complex to visualize in a manner that is it should be intuitive and accurate. This visualization method must preserve local and global structure in higher dimensional data, denoise the data so that the underlying structure is visible, and preserve as much information as possible i.e. local and global structure, in low dimensions (two to three). In addition, a visualization method should be robust in the sense that the obtained data structure is insensitive to the user configurations of the algorithm and scalable to the massive sizes of contemporary data. Potential of Heat-diffusion for Affinity-based Transition Embedding (PHATE) [16] is designed for these objectives.

There are three main steps of the algorithm. The first step is to use local similarities to encode local data information. The second step is to use potential distances to encode global relationships in data. The third one is to have low-dimensional data by embedding potential distance information.

III. METHODOLOGY

The research has been divided into two phases. Feature selection is the first phase in which different clustering techniques are used. These selected features are analyzed against five datasets. Along with the clustering techniques, some

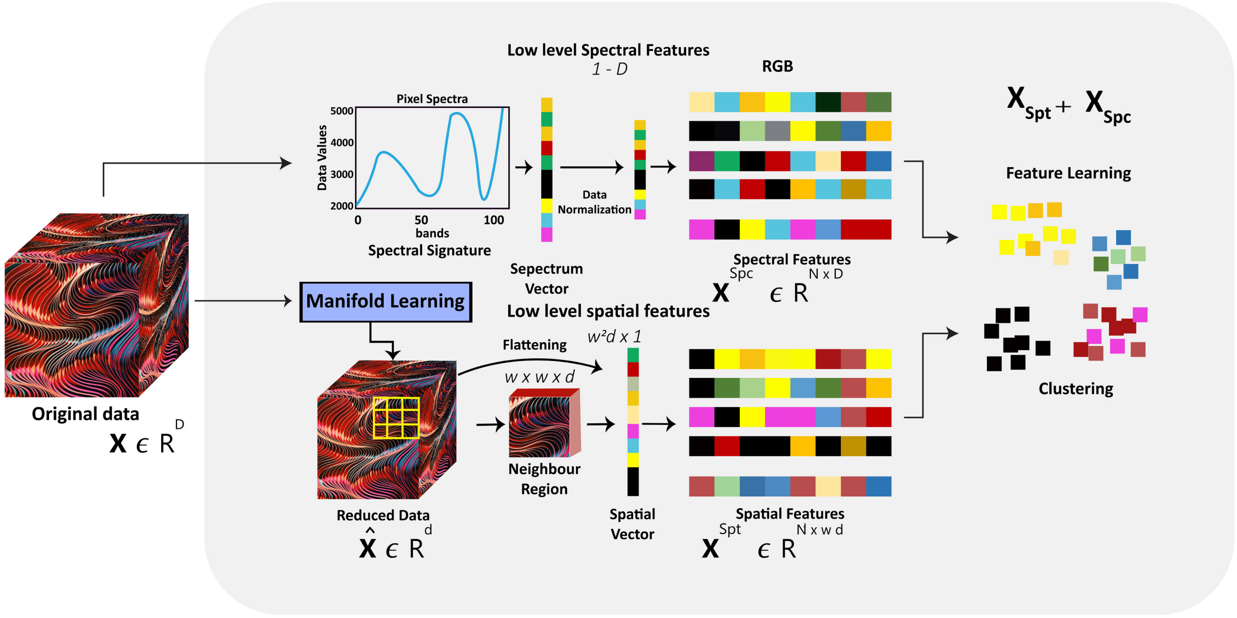


FIGURE 2. Adaptive feature selection includes techniques such as feature extractor and clustering algorithms for feature selection.

manifold learning techniques are hybridized in attention to attain better performances. In the following sections, the manifold learning techniques and the clustering techniques are explained respectively. The second phase of the research is about image classification, where a new model TLCA is proposed and the three state-of-the-art algorithms are tested and evaluated for image classification.

A. ADAPTIVE FEATURE SELECTION

The objective of feature selection for clustering is to select a set of most relevant features that facilitate the discovery of natural clusters in the data, according to the selected criterion [17]. These selected features may lead to the best version of relevant features if a suitable feature extraction technique is applied to consider the spatial features of image data X. Fig. 2 represents the complete flow of how data spectrum is used to capture spectral features and how feature extraction is performed. As normal preprocessing steps, data normalization and data scaling of a spectral signature are used to provide the spectral features of an image. Therefore, we add feature extraction using isomap, LLE, UMAP, or PHATE. These techniques are known as manifold learning techniques explained in Section II.

Clustering algorithms discussed in the literature are sensitive to largeness or dimensionality or both. There is an entropy-based solution is proposed for the ranking of features [18]. The key issue regarding this resolution is the repeated calculations required for the information-entropy-based significance of an attribute set, which slows down feature selection for large datasets. Consequently, feature extraction followed by feature selection assisted in this regard. Adaptive feature selection involves different combinations of clustering techniques and manifold learning techniques. Experimentally tested clustering methodologies are discussed below.

1) K-MEAN

It works in an iterative process [19], of assigning all the data points to the groups with the initial supposition of a specific centroid to each cluster. This assignment of data points is done by calculating the Euclidean distance (8) between the data points and the supposed centroids.

$$\mathrm{d}\,(\mathrm{x},\mathrm{y}) = \sqrt{\sum\nolimits_{\mathrm{i}=1}^{\mathrm{n}} (\mathrm{x_i} - \mathrm{y_i})^2} \tag{8}$$

The centroid chosen for a fixed number of clusters in the first step keeps on changing to minimize the sum of distances between the data points and the assigned centroids.

$$C_i = \frac{1}{|N_i|} \sum X_i \tag{9}$$

2) SPECTRAL CLUSTERING

Numerous fields, such as data analysis, video indexing, character identification, image processing, speech separation, etc., have effectively implemented spectral clustering. In these applications and many more, the number of data elements to cluster can be extraordinarily large [20]. Basic concepts of spectral clustering involve algebraic graph theory and graph cut methods. The advanced development of spectral clustering comprises the aspects of similarity matrix, Laplacian matrix, selecting eigenvectors, and the number of clusters chosen. The main focus of spectral clustering is choosing a distance measurement that adequately describes the intrinsic structure of the data elements. Data within the same category should have a high degree of similarity and adhere to space consistency. The measurement of similarity is vital to the efficacy of spectral clustering [21]. As a rule, the Gaussian kernel function is chosen as the similarity measure.

Following the construction of a similarity matrix, the corresponding Laplacian matrix is created using various graph cut methods. The efficacy of spectral clustering algorithms is significantly influenced by the selection of graph cut methods and the construction of Laplacian matrices. Through eigen-decomposition, the eigenvalues and eigenvectors of a Laplacian matrix can be determined. An analysis of the properties of eigenspace demonstrates that: (a) not every Laplacian matrix's eigenvector is relevant for clustering; (b) eigenvector selection is crucial because using uninformative eigenvectors could result in poor clustering results; and (c) the corresponding eigenvalues cannot be used to select relevant eigenvectors for a realistic dataset.

3) GAUSSIAN MIXER MODEL

GMM [22] works the same as k-means does but k-means only performs better for the data distributed over circular shapes. The reason behind this is it clusters the points only in a circular shape with a radius defined by the most distant point. In the case of GMM, the clusters can be oblong depending upon the data distribution. Besides assigning a cluster to each point, GMM considers the probability that a certain point belongs to which cluster.

B. IMAGE CLASSIFICATION

A popular technique for classifying hyperspectral images is supervised classification. The fundamental procedure is to calculate the discriminant function and then establish the discriminant criterion based on the given sample category and prior knowledge; Support vector machine, artificial neural network (ANN) [23], convolutional neural network (CNN) [24], long short-term memory (LSTM), decision tree, gated recurrent unit networks (GRU) [25] and maximum likelihood classification methods are supervised classification techniques that are frequently employed. Some of these are described below.

1) CNN

CNN's structure includes the convolutional, pooling, nonlinear activation, and fully connected layers. In general, the image is preprocessed [26] before being provided to the network via the input layer, passed through a series of alternately arranged convolutional and pooling layers, and then a fully connected layer is used for classification.

CNN [27], [28] adds a very distinctive convolutional and pooling layer compared to Multilayer Perceptron (MLP). For large data sets, CNN exhibits exceptional cost performance in terms of model size, and its performance is better also. The convolutional layer has the properties of a local receptive field, which retains the input shape. Another point to be noticed is that the convolutional layer frequently calculates the same convolution kernel and various input positions through a sliding window, thereby effectively preventing the training parameter size from becoming excessively large. The pooling layer reduces the computational load by minimizing the number of connections between the convolutional layers [29] and alleviates the convolutional layer's excessive position sensitivity. CNN ensures the invariance of input image pixels with respect to displacement, scaling, and distortion.

2) LSTM

Long Short-Term Memory (LSTM) is a sophisticated form of Recurrent Neural Networks (RNN) that captures long-term dependencies. LSTM was introduced in 1997 [30] and improved in 2013 [31], garnering a great deal of popularity in the deep learning community. LSTM models have proven more effective than standard RNNs at retaining and utilizing information over extended sequences [32].

In an LSTM network, the current input at a particular time step and the output from the previous time step are supplied into the LSTM unit, which in turn generates an output that is passed on to the subsequent time step. Commonly, the final hidden layer of the last time phase, and sometimes all hidden layers, are used for classification purposes [33].

Three gates comprise LSTM: input gate, forget gate, and output gate. Each gate serves a distinct purpose in regulating the passage of information. Based on the current input and the preceding internal state, the input gate determines how to update the internal state. The forget gate determines how much of the preceding state of the internal environment should be forgotten. Lastly, the output gate regulates the effect of the system's internal state [34].

3) GRU

A gated recurrent unit (GRU) is an improvement on the conventional RNN (recurrent neural network). In 2014, Kyunghyun Cho [35] introduced it for statistical machine translation. More or less they are similar to LSTM. GRU also employs gates to control the information flow, just like LSTM. They are comparatively more recent than LSTM and are superior to LSTM in terms of simplicity of architecture.

Unlike LSTM, it lacks a distinct cell state (Ct) and possesses only a hidden state (Ht). Due to their simplified architecture, GRUs can be trained more quickly. Only two gates comprise GRU: Reset gate and Update gate. Equations for their functionalities are as follows.

$$r_t = \sigma(x_t \times U_r + H_{t-1} \times W_r) \tag{10}$$

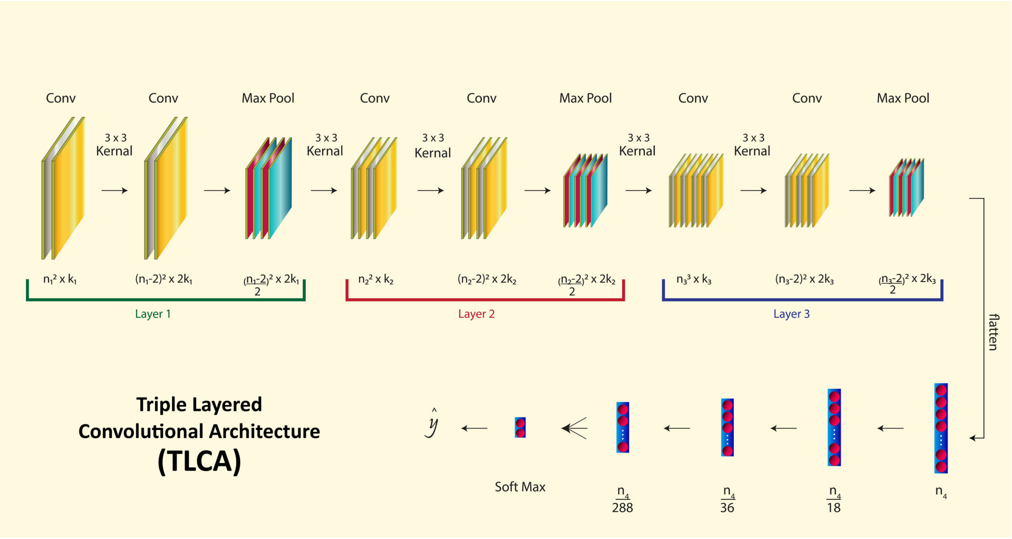


FIGURE 3. Proposed model of image classification.

TABLE 1. Model summary.

Layer (type)	Output Shape	Param #
conv2d_12 (Conv2D)	(None, 222, 222, 32)	890
conv2d_13 (Conv2D)	(None, 220, 220, 64)	18496
max_pooling2d_6 (MaxPooling2D)	(None, 110, 110, 64)	0
conv2d_14 (Conv2D)	(None, 108, 108, 64)	36928
conv2d_15 (Conv2D)	(None, 106, 106, 128)	73856
max_pooling2d_7 (MaxPooling 2D)	(None, 53, 53, 128)	0
conv2d_16 (Conv2D)	(None, 51, 51, 256)	295168
conv2d_17 (Conv2D)	(None, 49, 49, 128)	295040
max_pooling2d_8 (MaxPooling 2D)	(None, 24, 24, 128)	0
dropout_6 (Dropout)	(None, 24, 24, 128)	0
flatten_2 (Flatten)	(None, 73728)	0
dense_8 (Dense)	(None, 1024)	75498496
dropout_7 (Dropout)	(None, 1024)	0
dense_9 (Dense)	(None, 512)	524800
dropout_8 (Dropout)	(None, 512)	0
dense_10 (Dense)	(None, 64)	32832
dense_11 (Dense)	(None, 2)	3445

Total params: 76,779,957
Trainable params: 76,779,957
Non-trainable params: 0

$$u_t = \sigma(x_t \times U_u + H_{t-1} \times W_u) \tag{11}$$

The reset gate uses equation 1, where Ur and Wr are the weight matrices for the reset gate. Similarly, the update gate uses equation 2, where Uu and Wu are the weight matrices for update gate.

4) TLCA (PROPOSED MODEL)

TLCA has proven to be an effective solution for image classification problems. The efficacy of large image databases, such as the Pistachio, HAR, Mango Leaves, and Cards datasets have been significantly enhanced by the TLCA-based network. As an improved form of CNN, it is very adept at understanding the local and global structures from image data.

The overall design of the framework can be depicted in Fig. 3. The first part of layer 1 is a convolutional layer with 32 output channels and a kernel dimension of 3 × 3 pixels. The second part of layer 1 is also a convolutional layer with 64 output channels and the same kernel size. The third part of layer 1 is a max pooling layer with a 2 × 2 kernel. In a triple-layered architecture, the same sequence is repeated three times. Each of the subsequent five layers is composed of 73728-1024-512-64-c neurons and is fully connected.

Where ‘c’ is different for different datasets and is the number of classes each dataset has. Since the input image is not textual, the network must learn large-scale or high-level features. The network with a three-layered architecture performs image classification tasks significantly well. The large number of parameters to be learned may result in over-fitting, but as a consequence, accuracy improves. The results obtained using epochs 20 on batch size 32 are satisfactory. Model summary is shown in Table 1.

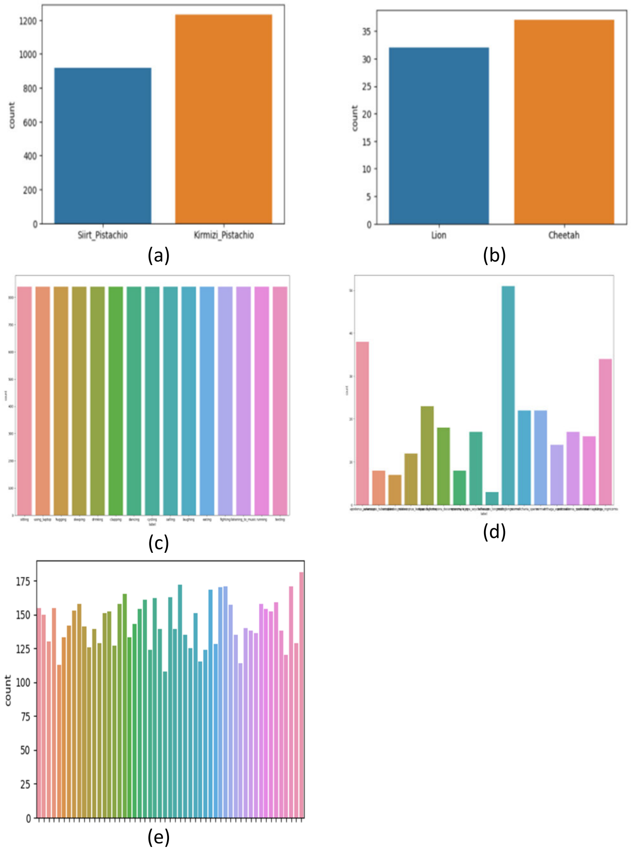
C. EXPERIMENTAL SETUP

To evaluate the proposed model of adaptive feature selection and image classification model TLCA, we used the following experimental setup and five datasets, whose prescription is mentioned as follows.

The experiment setup involves disk storage, system RAM, and GPU RAM as hardware requirements and Python3 as software prerequisites. Depending on dataset size and model

TABLE 2. Dataset description.

Sr. no	Dataset Name	Classes	Image count	Train/Test/Validate
(a)	Pistachio_dataset	2	2,147	1,610/322/215
(b)	Animal_dataset	2	69	51/11/7
(c)	HAR_dataset	15	28,992	21,744/4,348/2,899
(d)	Mango Leaves	16	14,200	10,650/2,130/1,420
(e)	Cards Image	53	7,624	5,718/1,144/762



(a)

(b)

(c)

(d)

(e)

FIGURE 4. Data distribution over classes.

complexity almost 32 GB of system RAM and 32 GB of disk storage is desired for data accumulation, model checkpoints, and other relevant files. We have used the Google Colab Pro+ version for our experimentation. A100 type of GPU has been chosen. The latest generation, A100-80GB doubles GPU memory and introduces the world's quickest memory bandwidth at 2 terabytes per second (TB/s), which accelerates time to solution for the largest models and largest datasets.

D. DATASETS

Five different image datasets mentioned in Table 2, have been taken from the Kaggle repository. The image data has been divided into 3 proportions for training, testing, and validation. Of the total images, 75% is used for training, 15% for testing, and 10% for validation. Their distribution can be seen in the table.

The distribution of data over different classes in each dataset is demonstrated in the histograms shown in Fig. 4. Datasets Pistachio and Animal are binary class datasets while others are multiclass. The dataset 'Human Action Recognition' (HAR) is a balanced dataset which means each class has an equal number of images and the dataset 'Mango Leaves' is the most imbalanced while others are near to balanced.

IV. EXPERIMENTAL RESULTS

A. ADAPTIVE FEATURE SELECTION

In the first phase of experimentation, three clustering algorithms; KMeans, GMM, and spectral clustering are used for feature selection. As expected the results are not satisfied so some manifold learning techniques: LLE, Isomap, UMAP, and PHATE are introduced as a preprocessing step for better performance. For different datasets, the different combination of clustering and manifold learning technique provides the best results. Three out of five datasets i.e. Animal, HAR, and Cards dataset perform better with PHATE + Kmeans while for the Pistachio dataset, its Isomap + Kmeans performs well. As far as simple clustering is concerned, spectral behaves a way better than Kmeans and GMM for this dataset but when these clustering techniques combine with Isomap, Kmeans provide better features. There are two most prominent cases where clustering accuracy improved remarkably by introducing manifold learning. For dataset HAR, it rises from 8.07% to 55.07% while for Cards, it elevates from 20.80% to 31.78%.

B. ADAPTIVE FEATURE SELECTION

In the second phase of experimentation, the proposed image classification model; TLCA is evaluated based on accuracy. Its performance is compared with the state-of-the-art image classification models CNN, LSTM, and GRU. Simple classification can be further improved by reducing data size before processing.

This data size reduction is dimension reduction which prevents overfitting and eliminates noise and redundancy. Eventually, the computational cost is reduced and generalized performance improves. As shown in Table 3, the accuracy level of TLCA for datasets Pistachio, Animal, and Cards is far better than CNN, LSTM, and GRU. The accuracy is further improved when feature extraction by PHATE is done and the reduced features are used for classification by TLCA. For Pistachio it goes from 85.34% to 97.73%. For HAR, it improves from 71.43% to 97.97% and for dataset Cards, the accuracy rises from 71.71% to 95.65%. The results of TLCA with PHATE i.e. Hybrid Triple Layered Convolutional Architecture (HTLCA) are also mentioned in the table.

To show the performance of TLCA, we presented the convergence graphs. Training accuracy and validation accuracy

TABLE 3. Adaptive feature selection results.

Dataset	Clustering Techniques	Simple Clustering	Clustering with Manifold Learning			
			LLE	Isomap	UMAP	PHATE
Pistachio	KMeans	27.88%	56.37%	**64.71%**	46.32%	59.21%
	GMM	30.67%	52.54%	31.79%	47.39%	60.24%
	Spectral Clustering	59.17%	57.54%	37.52%	50.54%	45.85%
Animal	KMeans	52.17%	50.52%	56.52%	55.07%	**56.53%**
	GMM	52.37%	50.72%	57.97%	55.24%	50.72%
	Spectral Clustering	50.72%	52.72%	43.47%	53.62%	50.71%
HAR	KMeans	07.19%	50.72%	43.47%	49.27%	**55.07%**
	GMM	06.76%	50.73%	42.02%	42.02%	50.72%
	Spectral Clustering	08.07%	47.82%	43.47%	45.45%	50.70%
Mango Leaves	KMeans	02.08%	02.55%	01.67%	01.68%	03.17%
	GMM	01.80%	01.38%	01.47%	01.85%	01.79%
	Spectral Clustering	1.49%	01.15%	01.43%	02.25%	01.47%
Cards	KMeans	20.80%	25.51%	16.74%	16.87%	**31.78%**
	GMM	10.80%	13.86%	14.78%	18.57%	17.92%
	Spectral Clustering	14.09%	11.51%	14.39%	22.50%	14.78%

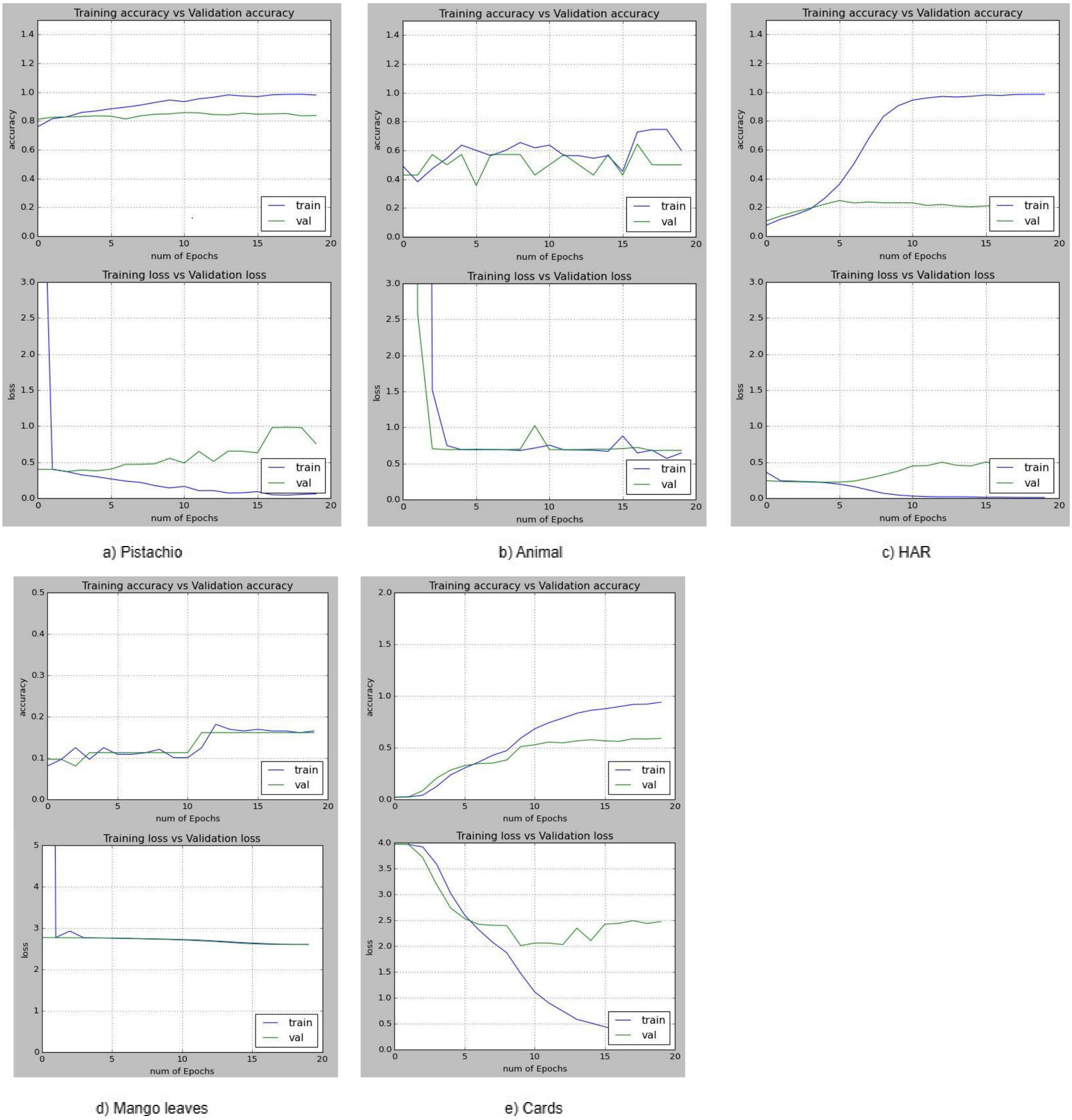
TABLE 4. Image classification results.

Dataset	Technique	Accuracy
Pistachio	CNN	83.72%
	LSTM	57.18%
	GRU	58.44%
	TLCA	**85.34%**
	HTLCA	**97.73%**
Animal	CNN	57.14%
	LSTM	52.73%
	GRU	57.11%
	TLCA	**58.14%**
	HTLCA	**60.18%**
HAR	CNN	**78.18%**
	LSTM	57.45%
	GRU	58.44%
	TLCA	71.43%
	HTLCA	**97.97%**
Images of Mango Leaves	CNN	29.03%
	LSTM	16.94%
	GRU	44.35%
	TLCA	25.81%
	HTLCA	16.53%
Cards Image Dataset	CNN	20.80%
	LSTM	10.92%
	GRU	11.07%
	TLCA	**71.71%**
	HTLCA	**95.65%**

curves are demonstrated in the upper half of Fig 4 for each of the 5 datasets. Similarly, training loss and validation loss curves are shown in the lower half of each figure. It can be observed from the accuracy plots of three datasets that accuracy is inclined over 20 epochs and the model performed well against state-of-the-art algorithms while convergence in loss plots is not that eminent, hence providing room for over-fitting to be handled. In HTLCA, we first extract features then reduce dimensions using PHATE, and then train TLCA offline using the labeled image dataset.

V. DISCUSSION

Adaptive feature learning via clustering is introduced in the first phase of this research. K-means, GMM, and spectral clustering It is evident that feature selection can be significantly improved using manifold learning techniques. In the second phase of experimentation, the proposed image classification model HTLCA is tested, and the accuracy is compared with state-of-the-art classification models: CNN, LSTM and GRU. The association with manifold learning further improves classification performance. The convergence graph for the training/validation accuracy and loss of the proposed model shows how it behaves (Fig. 5); good fit in some cases while overfitting for other datasets. The results show that there is no overfitting in the case of small datasets. Among the model accuracy mentioned in Table 4, TLCA achieved the best classification performance for the Pistachio and Cards image datasets, with an accuracy of 85.34% and 71.71%. Using PHATE as a preprocessing step (HTLCA)



a) Pistachio

b) Animal

c) HAR

d) Mango leaves

e) Cards

FIGURE 5. Graph of convergence for HTLCA.

increases the classification accuracy up to 97.73%, 60.18%, 97.97%, and 95.65% for Pistachio, Animal, HAR, and Cards datasets respectively. Here are some useful insights observed in experimental results.

For Larger datasets Pistachio, HAR, Mango Leaves, and Cards, we observe a smooth curve of TLCA accuracy(Fig. 5), while for dataset 'Animal', jerks are found. This dataset is about 100 times smaller than others, so adequate training data is required for better model performance. For datasets Pistachio, Mango Leaves, and Animal, the training accuracy and validation accuracy graph lines are much closer to each other, which depicts a very small overfitting of data. It means the model is performing well on unseen data. Contrary to this, the data is overfitted for the datasets HAR and

Cards. Accuracy may be compromised for smaller datasets and the datasets where data distribution over classes is not balanced. For instance, the Animal dataset is small (Table 2), and for Mango Leaves, data distribution over 16 classes is unbalanced (see Fig. 4).

VI. CONCLUSION AND FUTURE WORK

Manifold learning is a technique of machine learning and data analysis that extracts significant features from high-dimensional data. Different clustering algorithms have different performances on various datasets for feature selection. Their accuracies can be enhanced using manifold learning techniques i.e. PHATE, UMAP, isomap, and LLE. The extracted features can also assist in image classification. Therefore, feature extraction by manifold learning followed by adaptive feature selection or image classification performs well and can be depicted by experimental results. Animal, HAR and Cards datasets perform better with PHATE followed by Kmeans while for the Pistachio dataset, its Isomap followed by Kmeans that performs well. In the second phase of experimentation, the proposed image classification model TLCA is evaluated compared with modern classification models: CNN, LSTM, and GRU and governed the accuracies of 97.73%, 60.18%, 97.97%, and 95.65% for Pistachio, Animal, HAR and Cards dataset respectively.

In the future, this research can be extended with dimension reduction by auto-encoders. As we see, in the results how drastically performance accelerated by using manifold learning techniques, extra feature reduction can cause lesser training times, enhancing or at least retaining the accuracy level of feature selection and image classification. Moreover, work can be done to resolve data overfitting issues.

DATA AVAILABLE STATEMENT

Code will be available on demand.

REFERENCES

[1] A. J. Izenman, "Introduction to manifold learning," *Wiley Interdiscip. Rev. Comput. Stat.*, vol. 4, no. 5, pp. 439–446, 2012, doi: 10.1002/wics.1222.

[2] E. Oja, "The nonlinear PCA learning rule in independent component analysis," *Neurocomputing*, vol. 17, pp. 25–45, Sep. 1997, doi: 10.1016/S0925-2312(97)00045-3.

[3] G. H. L. van der Maaten, "Visualizing data using t-SNE," *Ann. Oper. Res.*, vol. 219, no. 1, pp. 187–202, 2014, doi: 10.1007/s10479-011-0841-3.

[4] Y. Zhang, Z. Zhang, J. Qin, L. Zhang, B. Li, and F. Li, "Semi-supervised local multi-manifold isomap by linear embedding for feature extraction," *Pattern Recognit.*, vol. 76, pp. 662–678, Apr. 2018, doi: 10.1016/j.patcog.2017.09.043.

[5] D. Lunga, S. Prasad, M. M. Crawford, and O. Ersoy, "Manifold-learning-based feature extraction for classification of hyperspectral data: A review of advances in manifold learning," *IEEE Signal Process. Mag.*, vol. 31, no. 1, pp. 55–66, Jan. 2014, doi: 10.1109/MSP.2013.2279894.

[6] J. Zhang, S. Z. Li, and J. Wang, "Manifold learning and applications in recognition," in *Intelligent Multimedia Processing With Soft Computing*. Berlin, Germany: Springer, 2006, pp. 281–300, doi: 10.1007/3-540-32367-8_13.

[7] Y. Fan and Z. Zhao, "Cryo-electron microscopy image analysis using multi-frequency vector diffusion maps," 2019, *arXiv:1904.07772*.

[8] B. Liu, S.-X. Xia, F.-R. Meng, and Y. Zhou, "Manifold regularized extreme learning machine," *Neural Comput. Appl.*, vol. 27, no. 2, pp. 255–269, Feb. 2016, doi: 10.1007/s00521-014-1777-8.

[9] X. Chen, R. Chen, Q. Wu, F. Nie, M. Yang, and R. Mao, "Semisupervised feature selection via structured manifold learning," *IEEE Trans. Cybern.*, vol. 52, no. 7, pp. 5756–5766, Jul. 2022, doi: 10.1109/TCYB.2021.3052847.

[10] J. Li, K. Cheng, S. Wang, F. Morstatter, R. P. Trevino, J. Tang, and H. Liu, "Feature selection: A data perspective," *ACM Comput. Surv.*, vol. 50, no. 6, pp. 1–45, Nov. 2018, doi: 10.1145/3136625.

[11] S. Lv, S. Shi, H. Wang, and F. Li, "Semi-supervised multi-label feature selection with adaptive structure learning and manifold learning," *Knowl.-Based Syst.*, vol. 214, Feb. 2021, Art. no. 106757, doi: 10.1016/j.knosys.2021.106757.

[12] J. Alzubi, A. Nayyar, and A. Kumar, "Machine learning from theory to algorithms: An overview," *J. Phys., Conf.*, vol. 1142, Nov. 2018, Art. no. 012012, doi: 10.1088/1742-6596/1142/1/012012.

[13] H. Bhaskar, D. C. Hoyle, and S. Singh, "Machine learning in bioinformatics: A brief survey and recommendations for practitioners," *Comput. Biol. Med.*, vol. 36, no. 10, pp. 1104–1125, Oct. 2006, doi: 10.1016/j.compbiomed.2005.09.002.

[14] D. Liao, Y. Qian, and Y. Y. Tang, "Constrained manifold learning for hyperspectral imagery visualization," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 11, no. 4, pp. 1213–1226, Apr. 2018, doi: 10.1109/JSTARS.2017.2775644.

[15] N. Hoque, D. K. Bhattacharyya, and J. K. Kalita, "MIFS-ND: A mutual information-based feature selection method," *Exp. Syst. Appl.*, vol. 41, no. 14, pp. 6371–6385, Oct. 2014, doi: 10.1016/j.eswa.2014.04.019.

[16] K. R. Moon, D. van Dijk, Z. Wang, S. Gigante, D. B. Burkhardt, W. S. Chen, K. Yim, A. V. D. Elzen, M. J. Hirn, R. R. Coifman, N. B. Ivanova, G. Wolf, and S. Krishnaswamy, "Visualizing structure and transitions in high-dimensional biological data," *Nature Biotechnol.*, vol. 37, no. 12, pp. 1482–1492, Dec. 2019, doi: 10.1038/s41587-019-0336-3.

[17] J. R. Adhikary and M. N. Murty, "Feature selection for unsupervised learning," in *Neural Information Processing* (Lecture Notes in Computer Science), vol. 7665. Berlin, Germany: Springer, 2012, pp. 382–389, doi: 10.1007/978-3-642-34487-9_47.

[18] M. Dash and H. Liu, "Feature selection for clustering," in *Knowledge Discovery and Data Mining, Current Issues and New Applications* (Lecture Notes in Computer Science), vol. 1805. Berlin, Germany: Springer, 2000, pp. 110–121, doi: 10.1007/3-540-45571-x_13.

[19] J. Yadav and M. Sharma, "A review of K-mean algorithm," *Int. J. Eng. Trends Technol.*, vol. 4, no. 7, pp. 2972–2976, 2013.

[20] H. Jia, S. Ding, X. Xu, and R. Nie, "The latest research progress on spectral clustering," *Neural Comput. Appl.*, vol. 24, nos. 7–8, pp. 1477–1486, Jun. 2014, doi: 10.1007/s00521-013-1439-2.

[21] L. Wang, L. F. Bo, and L. C. Jiao, "Density-sensitive spectral clustering," *Acta Electron. Sin.*, vol. 35, no. 8, pp. 1577–1581, 2007.

[22] G. J. McLachlan and S. Rathnayake, "On the number of components in a Gaussian mixture model," *WIREs Data Mining Knowl. Discovery*, vol. 4, no. 5, pp. 341–355, Sep. 2014, doi: 10.1002/widm.1135.

[23] F. Paquin, J. Rivnay, A. Salleo, N. Stingelin, and C. Silva, "Multi-phase semicrystalline microstructures drive exciton dissociation in neat plastic semiconductors," *J. Mater. Chem. C*, vol. 3, pp. 10715–10722, Jan. 2015, doi: 10.1039/C5TC02043C.

[24] D. Bhatt, C. Patel, H. Talsania, J. Patel, R. Vaghela, S. Pandya, K. Modi, and H. Ghayvat, "CNN variants for computer vision: History, architecture, application, challenges and future scope," *Electronics*, vol. 10, no. 20, p. 2470, Oct. 2021, doi: 10.3390/electronics10202470.

[25] R. Dey and F. M. Salem, "Gate-variants of gated recurrent unit (GRU) neural networks," in *Proc. IEEE 60th Int. Midwest Symp. Circuits Syst. (MWSCAS)*, Aug. 2017, pp. 1597–1600, doi: 10.1109/MWSCAS.2017.8053243.

[26] D. Ciregan, U. Meier, and J. Schmidhuber, "Multi-column deep neural networks for image classification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2012, pp. 3642–3649, doi: 10.1109/CVPR.2012.6248110.

[27] M. Oquab, L. Bottou, I. Laptev, and J. Sivic, "Learning and transferring mid-level image representations using convolutional neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 1717–1724, doi: 10.1109/CVPR.2014.222.

[28] I. H. Md Yusof, M. An, and M. H. Barghi, "Integration of lean construction considerations into design process of construction projects," in *Proc. 31st Annu. Assoc. Res. Constr. Manag. Conf.*, 2015, pp. 885–894.

[29] J. Gu, Z. Wang, J. Kuen, L. Ma, A. Shahroudy, B. Shuai, T. Liu, X. Wang, G. Wang, J. Cai, and T. Chen, "Recent advances in convolutional neural networks," *Pattern Recognit.*, vol. 77, pp. 354–377, May 2018, doi: 10.1016/j.patcog.2017.10.013.

[30] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997, doi: 10.1162/neco.1997.9.8.1735.

[31] A. Graves, "Generating sequences with recurrent neural networks," 2013, *arXiv:1308.0850*.

[32] F. M. Shiri, T. Perumal, N. Mustapha, R. Mohamed, M. A. B. Ahmadon, and S. Yamaguchi, "A survey on multi-resident activity recognition in smart environments," 2023, *arXiv:2304.12304*.

[33] S. Minaee, E. Azimi, and A. Abdolrashidi, "Deep-sentiment: Sentiment analysis using ensemble of CNN and bi-LSTM models," 2019, *arXiv:1904.04206*.

[34] W. Fang, Y. Chen, and Q. Xue, "Survey on research of RNN-based spatio-temporal sequence prediction algorithms," *J. Big Data*, vol. 3, no. 3, pp. 97–110, 2021, doi: 10.32604/jbd.2021.016993.

[35] K. Cho, B. van Merrienboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using RNN encoder–decoder for statistical machine translation," in *Proc. Conf. Empirical Methods Natural Lang. Process. (EMNLP)*, 2014, pp. 1724–1734, doi: 10.3115/v1/d14-1179.

AMNA ASHRAF was born in Bahawalpur, Pakistan, in 1992. She received the bachelor's and master's degrees in computer engineering from the University of Engineering and Technology, Lahore, Pakistan, in 2014 and 2017, respectively. She is currently pursuing the Ph.D. degree with Universiti Tun Hussein Onn Malaysia (UTHM) under the kind supervision of Prof. Nazri Mohd Nawi and Muhammad Aamir.

She is working on dimensionality reduction of large image datasets. Her research interest includes artificial intelligence. Another embryonic interest is to deal with the hyperspectral images from satellites.

NAZRI MOHD NAWI received the bachelor's degree from Universiti Sains Malaysia (USM), the master's degree in computer science from University Teknologi Malaysia (UTM), and the Ph.D. degree in data mining from Swansea University, Wales, U.K.

He is currently a Professor with the Department of Software Engineering, Faculty of Computer Science and Information Technology, Universiti Tun Hussein Onn Malaysia (UTHM), where he has been a Faculty Member, since 2001. In recent years, he has focused on better techniques for classification, analyzing, and hybridizing some new improvements on ANN using meta-heuristic techniques. He has successfully supervised a few Ph.D. students and currently, he is supervising eight Ph.D. students and published more than 100 papers in journals and conference proceedings. His research interests include soft computing and data mining techniques, particularly in artificial neural networks, ranging from theory to design and implementation. He has been involved with many conferences and workshop program committees and serves as a reviewer for many outstanding journals and international conferences.

MUHAMMAD AAMIR received the master's degree in computer science from the City University of Science and Information Technology, Pakistan, and the Ph.D. degree in information technology from University Tunn Hussien Onn Malaysia, in 2020.

Since October 2020, he has been a Research Data Scientist and Machine Learning Development Engineer with the University of Derby, U.K. He was a Data Scientist with Xululabs LLC, for the past two years. A prolific contributor to academia, he has authored more than 25 journal articles and conference papers, in addition to being a coauthor of a book focused on data analysis. His research interests include data science, deep learning, and computer programming.