Label Propagation Algorithm for Face Clustering using Shared Nearest Neighbor Similarity

Gao Yousheng
College of Computing, Informatics and Mathematics, Universiti Teknologi MARA, Shah Alam, Selangor Malaysia | College of Information Engineering, Jiujiang Vocational University, Jiu Jiang, Jiang Xi, China
2022287318@student.uitm.edu.my

Raseeda Hamzah
Computing Sciences Studies, College of Computing, Informatics and Mathematics, Universiti Teknologi MARA (UiTM), Melaka Branch, Malaysia
raseeda@uitm.edu.my

Siti Khatijah Nor Abdul Rahim
College of Computing, Informatics and Mathematics, Universiti Teknologi MARA, Shah Alam, Selangor, Malaysia
sitik781@uitm.edu.my

Raihah Aminuddin
Computing Sciences Studies, College of Computing, Informatics and Mathematics, Universiti Teknologi MARA (UiTM), Melaka Branch, Malaysia
raihah1@uitm.edu.my (corresponding author)

Li Ang
College of Computing, Informatics and Mathematics, Universiti Teknologi MARA, Shah Alam, Selangor, Malaysia | College of Information Engineering, Jiujiang Vocational University, Jiu Jiang, Jiang, Xi, China
2022667284@student.uitm.edu.my

Received: 2 September 2024 | Revised: 3 October 2024 and 11 October 2024 | Accepted: 13 October 2024

ABSTRACT

Facial image datasets are particularly vulnerable to challenges such as lighting variations and occlusion, which can complicate data classification. Semi-supervised learning, using a limited amount of labeled facial data, offers a solution by enhancing face classification accuracy while reducing manual labeling efforts. The Label Propagation Algorithm (LPA) is a commonly used semi-supervised algorithm that employs Radial Basis Function (RBF) to measure similarities between data nodes. However, RBF struggles to capture complex nonlinear relationships in facial data. To address this, an improved LPA is proposed that integrates Shared Nearest Neighbor (SNN) to enhance the correlation measurement between facial data and RBF. Three known datasets were considered: FERET, Yale, and ORL. The experiments showed that in the case of insufficient label samples, the accuracy reached 89.76%, 92.46%, and 81.48%, respectively. The proposed LPA enhances clustering robustness by introducing 128 dimensional facial features and more complex similarity measurement. The parameter of similarity measurement can be adjusted based on the characteristics of different datasets to achieve better clustering results. The improved LPA achieved better performance and face clustering effectiveness by enhancing robustness and adaptability.

Keywords-machine learning; label propagation algorithm; k-means; pairwise constraints; shared nearest neighbor

I. INTRODUCTION

The development of machine learning technology has greatly improved the performance of facial recognition models [1], but in the face of complex real-world scenarios, large-scale annotated data remains one of the main factors determining its performance limit [2]. Machine learning methods can be divided into three categories: unsupervised, supervised, and semi-supervised learning, based on whether the method uses label information. Collecting a large amount of labeled data is challenging in practical applications, which is why semi-supervised learning methods have gained significant attention [3, 4]. Semi-supervised learning makes full use of both labeled and unlabeled samples and incorporates both training data and test data during the training process [5, 6]. Therefore, it can use more information, such as the distribution characteristics of data. Semi-supervised Learning can achieve better learning results when the total data volume is large and the number of label data is relatively small [7, 8]. Therefore, it has received widespread attention. Semi-supervised classification methods can utilize a large amount of unlabeled data to guide classification [9], reducing data annotation while improving classification performance [10]. The Label Propagation Algorithm (LPA) performs semi-supervised classification by transferring labels based on similarities between the labeled and the unlabeled data [11]. The labels are assigned to unlabeled data based on category probability. Semi-supervised learning methods can be divided into five categories: production model, self-training, collaborative training, maximum separation, and graph-based. Among them, graph-based methods now consist one of the most concerned research directions in machine learning research due to their fast computing speed and high accuracy [12]. Graph-based semi-supervised learning mainly focuses on the construction of graphs [13], so building a high-quality graph is a key issue.

Shared Nearest Neighbor (SNN) algorithm is an improved version of the k-Nearest Neighbor (kNN) algorithm [14]. The SNN idea is to first construct a similarity matrix, then sparsely process the nearest k neighbors [15], and use this to construct a nearest neighbor graph, so that only samples with strong connections have links [16]. Then, the link strength of all sample points is calculated, and stronger sample points are more likely to be clustered into one class [17]. The advantage of SNN algorithm is that it can be applied to datasets with different densities and shapes, and can deal with data sets with large density differences [18].

Face clustering is a complex task because faces can exhibit significant changes under different conditions [19]. In different poses, occlusion, and lighting conditions, the facial features of the same person may appear completely different, making recognition and clustering more difficult [20]. Due to significant changes in posture, occlusion, lighting, and number of instances, face clusters exhibit significant differences in size, shape, and density [21]. Graph-based semi-supervised learning only considers the relationship between samples and adjacent samples when constructing the connection graph of the global structure [22], ignoring the possible imbalance problem of each category of samples. Based on the above, this paper proposes a face recognition label propagation algorithm based on the SNN similarity matrix. This algorithm considers the distribution environment of face data, and improves the performance of the original algorithm in processing face data sets of different sizes, shapes and densities.

II. METHOD

This section explains the basic working principles of SNN and LPA. When facing face clustering tasks, after analyzing the shortcomings of LPA, an improved version of the LPA algorithm by combining SNN and Radial Basis Function (RBF) is proposed. The workflow is described in Figure 1, which shows the flowchart of the LPA and experimental method applied. At the beginning, LPA, pairwise constraints, similarity matrix, and SNN are introduced. Then, the existing issues of LPA are analyzed. To overcome the LPA shortcomings, the LPA based on Shared Nearest Neighbor Similarity Matrix (LPASNNSM) is proposed. Accuracy is selected as the evaluation indicator. Finally, the performance of the algorithm was experimentally verified on one artificial dataset and three public face datasets.

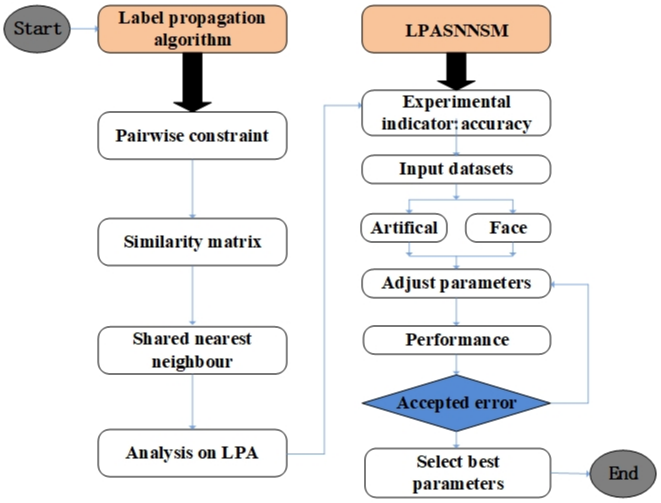

Fig. 1. Research flow.

A. *Label Propagation Algorithm*

The process of LPA can be described as follows:

- Input:

Labeled sample sets:

$$D_l = \{(x_1, y_1), (x_2, y_2), ..., (x_m, y_m)\} \subset R^N$$

Unlabeled sample sets:

$$D_u = \{x_1 x_2, ..., x_n\} \subset R^N .$$

Sample sets: $L = \{1, 2, ..., c\}$

Loop variable $i = 0$, initialize $y_u^{(i)} = 0$.

- Output:

Labeled sample sets $D_l' = \{x_1 x_2, ..., x_n\} \subset R^N$.

- Steps

1. LPA is based on graphs, so it is necessary to first build a graph for all the data, where the node of the graph is a data point that contains labeled and unlabeled data. The edges of nodes *i* and *j* represent their similarity. There are generally two composition methods for LPA, one is the kNN composition and the other is the RBF composition [25]. The kNN composition only retains the k-nearest neighbor weight of each node, while the others are 0, which means there are no edges. When composing RBF, the similarity matrix is calculated by:

$$w_{ij} = \exp(-\frac{\| x_i - x_j \|^2}{2\varepsilon^2}), \forall i \neq j \qquad (1)$$

where w_{ij} represents the weight value of each point in the correlation matrix, ε is the composition parameter, determined according to the specific algorithm, and sample x_i belongs to the set of *k* nearest neighbor samples of sample x_j.

2. Calculate the transition probability matrix based on the known *W*:

$$P_{ij} = \frac{W_{ij}}{\sum_{k=1}^{n} W_{ik}} \qquad (2)$$

where P_{ij} represents the transition probability from node i to node j .

3. Calculate and update the probability distribution:

$$F_{ij} = \sum_{k=1}^{n} P_{ik} F_{kj} \qquad (3)$$

4. Repeat steps 2 and 3 until *F* converges.
5. Determine the category of unlabeled samples:

$$y_i = \arg\max_{m \leq c} F_{im}, \quad 1 \leq i \leq n \qquad (4)$$

F_{im} represents the probability of data category *m* in sample x_i , while replacing the original D_u' with the newly generated D_l .

B. Shared Nearest Neighbor

Most traditional methods directly use distance to define similarity, such as the closer the distance between two objects, the higher the similarity between them [26]. By using deep learning to extract features from facial images, high-dimensional facial data can be obtained [27]. However, traditional similarity in high-dimensional space cannot accurately measure the similarity between two points. In response to the above issues, it can be addressed by the indirect method of similarity SNN. The principle is that if the direct similarity measure between two points cannot reflect their similarity, the two points are still similar. The following explanation can be made for SNN:

x_i and x_j are any two points in the sample set $\{x_1, x_2, ..., x_n\}$. If two points belong in the k-nearest neighbor region of each other, then the two points are similar, and the number of shared nearest neighbor points is the similarity of the two points. Similarity is defined as follows:

$$Similarity\ (x_i, x_j) = size(nn[x_i] \cap nn[x_j]) \qquad (5)$$

where $nn[x_i]$ and $nn[x_j]$ are the nearest k-nearest neighbor lists of x_i and x_j , with *size*(*A*) representing the size of set *A* and <*V*, *E*> represents the SNN graph ∀ *u*, *v* ∈ *Ω*. There is a connection between *u* and *v* only if $u \in nn[v]$ and $v \in nn[u]$. The connection strength is calculated by (5).

The density of point x_i is the number of points in the k-nearest neighbor list that are similar to x_i :

$$Density(x_i) = count(Similarity(x_i, x_j)) \geq n) \qquad (6)$$

In the k-nearest neighbor list of x_j , *n* is the threshold for determining whether x_i and x_j are similar. The threshold is the condition for two points to be similar, that they share *n* or more than *n* nearest neighbors.

Figure 2 shows the SNN similarity between two black nodes. Calculate the 8 closest neighbors of the two black nodes, and the four gray nodes in the figure are the nearest neighbors shared by two black nodes. So the SNN similarity between these two black nodes is 4.

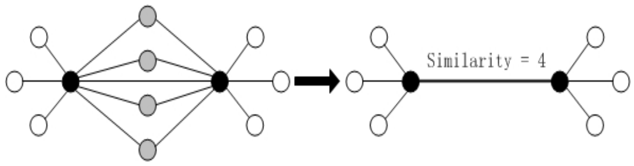

Fig. 2. SNN similarity between two nodes.

C. Label Propagation Algorithm based on Shared Nearest Neighbor Similarity Matrix

LPA is a graph based semi-supervised learning method. Its basic idea is to use the label information of labeled nodes to predict the label information of unlabeled nodes. The relationship between samples is used to establish a complete graph model. In a complete graph, nodes include labeled and unlabeled data, and their edges represent the similarity of two nodes. The labels of nodes are transferred to other nodes according to the similarity. Label data are like a source that can label unlabeled data. The greater the similarity of nodes, the easier it is for labels to propagate. Although LPA is simple and easy to implement and has good classification effect, its accuracy is not high when processing face data with large differences in shape and density. In order to improve the accuracy of LPA calculation when processing face image data, this paper proposes LPASNNSM, which is described below:

- Input: Labeled sample sets D_l and unlabeled sample sets D_u. Sample sets $L=\{1,2,...,c\}$, loop variable $i=0$, initialize $y_u^{(i)}=0$.
- Output: Labeled sample sets $D_l^{'}$.
- Steps:

1. Calculate the SNN similarity by:

$$SNN(x_i,x_j)=size\ (nn[x_i]\cap nn[x_j]) \qquad (7)$$

where $SNN(x_i,x_j)$ represents the number of neighbors shared by sample points x_i and x_j in the k-neighborhood, $nn[x_i]$ and $nn[x_j]$ are the nearest k-nearest neighbor lists of x_i and x_j, with *size*(*L*) representing the size of set *L*.

2. Combining SNN similarity, calculate the incidence matrix $w_{ij}^{n\times n}$:

$$w_{ij}=\left[1+\frac{2\delta}{1+\exp^{-SNN(x_i,y_j)}}\right]\cdot\exp(-\frac{\|x_i-x_j\|^2}{2\varepsilon^2}),\quad \forall i\neq j \; (8)$$

where w_{ij} represents the weight value of each point in the correlation matrix that combines SNN similarity, ε is the composition parameter, determined according to the specific algorithm, and sample x_j belongs to the set of *k* nearest neighbor samples of sample x_i.

3. Calculate the transition probability matrix P_{ij} based on the known *W* according to (2).
4. Calculate and update the probability distribution F_{ij} according to (3).
5. Repeat steps 3 and 4 until *F* converges.
6. Determine the category of unlabeled samples according to (4).

III. EXPERIMENT INDICATOR AND DATASETS

The experimental environment was an 11th Gen Intel (R) Core (TM) i5-11400H @ 2.70GHz, the memory was 24 GB DDR4 3200 Hz, and the programming environment was Python 3.8.0. The test was conducted on Windows 10 operating system.

To compare and analyze clustering results, the accuracy index was used. Accuracy is widely used to evaluate the performance of label propagation algorithms. The accuracy calculation equation is:

$$ACC=\frac{1}{n}\sum_{i=1}^{c}|T_i\cap P_i| \qquad (9)$$

where $T_i=\{T_1,T_2,...,T_n\}$ indicates that the original *n* data contain true *c* categories, and $P_i=\{P_1,P_2,...,P_n\}$ indicates *c* prediction categories of the *n* data after label propagation. T_i represents the number of points included in the *i*th category. P_i represents the data points contained in the *i*th category after label propagation, and $|P_i|$ represents the number of points contained in the set P. The accuracy index value is 0-1, and the larger the value is, the better the label propagation effect.

In this work, Dlib [28] was used to extract 128 dimensional face features. Dlib provides a pre-trained Convolutional Neural Network (CNN) model that can detect 68 key points on the face [29], covering different parts of the face such as eyes, eyebrows, nose, and mouth. Through the pre-trained face recognition models, Dlib can map facial images with 68 key points to a 128 dimensional feature vector. This 128 dimensional feature vector has good representational ability and can be used for tasks such as face recognition and face clustering.

To validate the proposed algorithm, one artificial dataset and three public face datasets were selected for this experiment. The experiment was conducted on one artificial dataset and three face datasets. The artificial dataset is generated from the make_circles package in the open-source machine learning library Scikit-learn [30]. The ORL face dataset can be accessed from the UCI Machine Learning Repository [31]. The Yale face dataset is available from Yale University's official website [32]. The FERET face dataset can be obtained from the National Institute of Standards and Technology (NIST) [33]. The artificial dataset consists of 1600 samples, with two concentric circles in the shape. Each circle represents a class. The ORL face dataset contains 4000 photos of 40 people. Each person has 10 photos, including face expressions, minor posture changes, and scale changes within 20%. The Yale face dataset contains 165 photos of 15 people, with 11 photos per person, mainly including changes in lighting conditions, face expressions, etc. The FERET face dataset contains 1400 images of 200 people. Each person has 7 pictures, including different postures, lighting conditions, etc. The datasets parameters are shown in Table I.

TABLE I. DATASET PARAMETERS

Datasets	Instances	Classes	Features
Circles	1600	2	2
Yale	165	15	128
ORL	4000	40	128
FERET	1400	200	128

IV. RESULTS AND DISCUSSION

Since RBF methods struggle to accurately capture complex nonlinear relationships in facial data, this paper introduces LPASNNSM to address this challenge. To verify the performance of LPASNNSM, the proposed article considered four datasets. Experiments were conducted on each dataset using kNN, RBF, and SNN to calculate similarity matrices. Kernel kNN indicates that the LPA uses kNN to calculate the similarity matrix. Kernel RBF represents calculating the similarity matrix according to (1). Kernel SNN represents calculating the similarity matrix according to (7). The kernel proposed in this paper computes the similarity matrix as detailed in (8).

Table II shows the experimental results on the Circles dataset. According to (1) and (8), the vale of ε of RBF and the proposed scheme was set to 0.2. After multiple experiments, setting δ of the proposed scheme to 0.1 achieved good results. In 10 iterations, the accuracy of kNN is 2.13% lower than that of RBF and the proposed method. SNN tends to converge after 10 iterations and kNN after 50, with an accuracy of nearly 67.77% remaining unchanged. After 100 iterations, the accuracy of the proposed model is about 1.5% to 4% higher than that of RBF. When the iteration number reached 162 rounds, the accuracy of the proposed method reached 100%, while the accuracy of kNN and RBF were 67.77% and 95.36%, respectively. Figure 3 shows the iterative process of running LPA on the Circle dataset through a two-dimensional diagram. Circles consist of a red circle and a blue circle. The red circle belongs to one category, while the blue circle belongs to another. At the beginning, only one red node and one blue node are marked. The colors of all nodes are accurately classified with the LPA algorithm.

TABLE II. CIRCLE DATASET EXPERIMENTAL RESULTS

Kernel	Iter10	Iter50	Iter100	Iter150	Iter162
kNN	53.94%	66.39%	67.77%	67.77%	67.77%
RBF	56.07%	82.16%	88.61%	93.55%	95.36%
SNN	50.00%	50.00%	50.00%	50.00%	50.00%
Proposed	56.07%	82.35%	90.17%	97.43%	100%

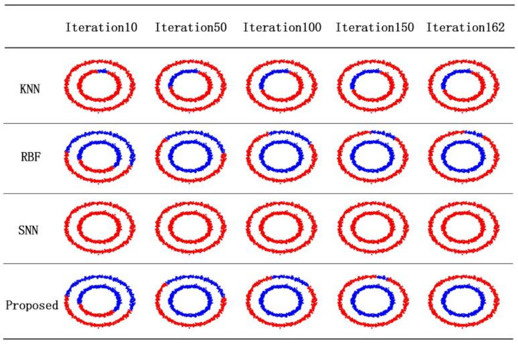

Fig. 3. Iteration process of running LPA on the Circle dataset.

Table III shows the experimental results on the FERET dataset. The ε of RBF and the proposed model was set to 0.06. After multiple experiments, setting δ of the proposed scheme to 0.6 achieved good results. In 100 iterations, the accuracy of kNN is 77.68%, which is lower than that of RBF and the proposed with accuracy of 87.83%, whereas kNN converged. After 200 rounds, the accuracy of the proposed model begins to improve compared to RBF. After 500 rounds, the proposed model converged with an accuracy of around 89.76%, while the accuracy of RBF was 89.51%.

TABLE III. FERET DATASET EXPERIMENTAL RESULTS

Kernel	Iter100	Iter200	Iter300	Iter400	Iter500
kNN	77.68%	77.68%	77.68%	77.68%	77.68%
RBF	87.83%	88.42%	88.75%	89.34%	89.51%
SNN	0.5%	0.5%	0.5%	0.5%	0.5%
Proposed	87.83%	88.50%	89.26%	89.68%	89.76%

Table IV shows the experimental results on the Yale dataset. The ε of RBF and the proposed model was set to 0.07. After multiple experiments, setting δ of the proposed scheme to 0.2 achieved good results. In 100 iterations, the accuracy of kNN was 73.97%, lower than that of RBF and proposed, whereas it has converged. RBF converged after 300 iterations. After 500 iterations, the proposed model converged with an accuracy of around 92.46%, while the accuracy of RBF was 88.35%.

TABLE IV. YALE DATASET EXPERIMENTAL RESULTS

Kernel	Iter100	Iter200	Iter300	Iter400	Iter500
kNN	73.97%	73.97%	73.97%	73.97%	73.97%
RBF	87.67%	87.67%	88.35%	88.35%	88.35%
SNN	6.12%	6.12%	6.12%	6.12%	6.12%
Proposed	88.35%	89.04%	91.09%	91.78%	92.46%

Table V shows the experimental results on the ORL dataset. The ε of RBF and SNN was set to 0.06 and the δ of the proposed model was set to 0.2. In 30 iterations, the accuracy of kNN was 65.81% lower than that of RBF and the proposed model, and it has converged. RBF converged after 100 rounds. The proposed model converged after 150 rounds with an accuracy of around 81.48%, while the accuracy of RBF was 79.20%.

TABLE V. ORL DATASET EXPERIMENT RESULTS

Kernel	Iter100	Iter200	Iter300	Iter400	Iter500
kNN	65.81%	65.81%	65.81%	65.81%	65.81%
RBF	77.20%	78.06%	79.20%	79.20%	79.20%
SNN	2.56%	2.56%	2.56%	2.56%	2.56%
Proposed	78.63%	79.20%	80.62%	80.91%	81.48%

The experimental results from different datasets offer a comprehensive view of the performance of kNN, RBF, SNN and the proposed algorithm. For Circles dataset, the proposed method excels in capturing complex patterns, achieving perfect accuracy which indicates it handles nonlinear relationships effectively. On the ORL dataset, the proposed method again demonstrates superior performance compared to kNN, RBF, and SNN especially after a higher number of iterations. On the FERET dataset, the proposed method provides slightly better accuracy than RBF, indicating its effectiveness. However, both methods perform similarly, with the proposed method achieving marginally better results. On the Yale dataset, the proposed method outperforms kNN, RBF, and SNN, achieving a higher accuracy, which suggests its robustness in handling different types of data. Overall, the proposed method consistently outperforms kNN and SNN and often surpasses RBF, particularly in more complex datasets and after a sufficient number of iterations. The proposed method generally converges to better accuracy levels compared to kNN, RBF, and SNN, which indicates it is better suited for datasets with intricate structures.

V. CONCLUSION

It is difficult for LPA to use RBF to capture accurate relationships between complex nonlinear face data. To address this issue, this article proposes semi-supervised LPA learning based on SNN. The algorithm introduces the concept of shared nearest neighbors between face samples, further enhancing the similarity between data based on RBF operation results. The final face dataset experiments verified the feasibility and higher classification accuracy of the algorithm proposed in this paper. The proposed LPA enhances clustering robustness by introducing 128 dimensional facial features and more complex similarity measurement. The parameter of similarity measurement can be adjusted based on the characteristics of different datasets to achieve better clustering results. Overall, the improved LPA can enhance the performance and effectiveness of face clustering by enhancing robustness and adaptability.

Although the algorithm in this article inherits the advantages of semi-supervised learning and improves the accuracy of facial image classification, due to the need to additionally strengthen the similarity between data, the computer memory demands are large, and the processing of higher dimensional face dataset is slow. These issues need to be further addressed in future research.

ACKNOWLEDGEMENT

The authors express gratitude to the Ministry of Higher Education Malaysia (MOHE) for their financial support under the Fundamental Research Grant Scheme (FRGS) (600-RMC/FRGS 5/3 (127/2024)) for the project titled "Robust Face Annotation Model through Modified Label Propagation Algorithm by Integrating Shared Nearest Neighbor and Kernel-Based Similarity,". We extend our gratitude to the College of Computing, Informatics and Mathematics at Universiti Teknologi MARA Melaka, CyberAI and Emerging Technology (CAIETRG) and Innovative Technology (INNOTECH) research group for their unwavering support.

REFERENCES

[1] G. Lou and H. Shi, "Face image recognition based on convolutional neural network," *China Communications*, vol. 17, no. 2, pp. 117–124, Feb. 2020, https://doi.org/10.23919/JCC.2020.02.010.

[2] N. F. Greenwald *et al.*, "Whole-cell segmentation of tissue images with human-level performance using large-scale data annotation and deep learning," *Nature Biotechnology*, vol. 40, no. 4, pp. 555–565, Apr. 2022, https://doi.org/10.1038/s41587-021-01094-0.

[3] X. Yang, Z. Song, I. King, and Z. Xu, "A Survey on Deep Semi-Supervised Learning," *IEEE Transactions on Knowledge and Data Engineering*, vol. 35, no. 9, pp. 8934–8954, Sep. 2023, https://doi.org/10.1109/TKDE.2022.3220219.

[4] W. Liu, H. Wang, X. Shen, and I. W. Tsang, "The Emerging Trends of Multi-Label Learning," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 11, pp. 7955–7974, Aug. 2022, https://doi.org/10.1109/TPAMI.2021.3119334.

[5] X. Wang, L. Lian, and S. X. Yu, "Unsupervised Selective Labeling for More Effective Semi-supervised Learning," in *European Conference on Computer Vision*, Tel Aviv, Israel, Oct. 2022, pp. 427–445, https://doi.org/10.1007/978-3-031-20056-4_25.

[6] P. Cascante-Bonilla, F. Tan, Y. Qi, and V. Ordonez, "Curriculum Labeling: Revisiting Pseudo-Labeling for Semi-Supervised Learning," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, no. 8, pp. 6912–6920, May 2021, https://doi.org/10.1609/aaai.v35i8.16852.

[7] M. Gao, Z. Zhang, G. Yu, S. Ö. Arık, L. S. Davis, and T. Pfister, "Consistency-Based Semi-supervised Active Learning: Towards Minimizing Labeling Cost," in *European Conference on Computer Vision*, Glasgow, United Kingdom, Aug. 2020, pp. 510–526, https://doi.org/10.1007/978-3-030-58607-2_30.

[8] S. Calderon-Ramirez *et al.*, "Dealing with Scarce Labelled Data: Semi-supervised Deep Learning with Mix Match for Covid-19 Detection Using Chest X-ray Images," in *25th International Conference on Pattern Recognition*, Milan, Italy, Jan. 2021, pp. 5294–5301, https://doi.org/10.1109/ICPR48806.2021.9412946.

[9] M. Han, H. Wu, Z. Chen, M. Li, and X. Zhang, "A survey of multi-label classification based on supervised and semi-supervised learning," *International Journal of Machine Learning and Cybernetics*, vol. 14, no. 3, pp. 697–724, Mar. 2023, https://doi.org/10.1007/s13042-022-01658-9.

[10] N. Mamat, M. F. Othman, R. Abdulghafor, A. A. Alwan, and Y. Gulzar, "Enhancing Image Annotation Technique of Fruit Classification Using a Deep Learning Approach," *Sustainability*, vol. 15, no. 2, Jan. 2023, Art. no. 901, https://doi.org/10.3390/su15020901.

[11] Q. Wang, C. Wang, H. Tang, D. Wu, and F. Wang, "Semi-supervised deep learning based on label propagation algorithm for debris flow susceptibility assessment in few-label scenarios," *Stochastic Environmental Research and Risk Assessment*, vol. 38, no. 7, pp. 2875–2890, Jul. 2024, https://doi.org/10.1007/s00477-024-02719-x.

[12] M. Baradaran and R. Bergevin, "A critical study on the recent deep learning based semi-supervised video anomaly detection methods," *Multimedia Tools and Applications*, vol. 83, no. 9, pp. 27761–27807, Mar. 2024, https://doi.org/10.1007/s11042-023-16445-z.

[13] C. Chen *et al.*, "Interactive Graph Construction for Graph-Based Semi-Supervised Learning," *IEEE transactions on visualization and computer graphics*, vol. 27, no. 9, pp. 3701–3716, Sep. 2021, https://doi.org/10.1109/TVCG.2021.3084694.

[14] T. Fan, Z. Yao, L. Han, B. Liu, and L. Lv, "Density peaks clustering based on k-nearest neighbors sharing," *Concurrency and Computation: Practice and Experience*, vol. 33, no. 5, 2021, Art. no. e5993, https://doi.org/10.1002/cpe.5993.

[15] Q. Liu, J. Yang, M. Deng, C. Song, and W. Liu, "SNN_flow: a shared nearest-neighbor-based clustering method for inhomogeneous origin-destination flows," *International Journal of Geographical Information Science*, vol. 36, no. 2, pp. 253–279, Feb. 2022, https://doi.org/10.1080/13658816.2021.1899184.

[16] L. Sun, X. Qin, W. Ding, J. Xu, and S. Zhang, "Density peaks clustering based on k-nearest neighbors and self-recommendation," *International Journal of Machine Learning and Cybernetics*, vol. 12, no. 7, pp. 1913–1938, Jul. 2021, https://doi.org/10.1007/s13042-021-01284-x.

[17] M. Yu and R. Cui, "Application of Digital Mining Facing Information Fusion Technology in the Field of National Costume Culture Design," *Mobile Information Systems*, vol. 2021, no. 1, 2021, Art. no. 3790413, https://doi.org/10.1155/2021/3790413.

[18] D. Hu, K. Liang, S. Zhou, W. Tu, M. Liu, and X. Liu, "scDFC: A deep fusion clustering method for single-cell RNA-seq data," *Briefings in Bioinformatics*, vol. 24, no. 4, Jul. 2023, Art. no. bbad216, https://doi.org/10.1093/bib/bbad216.

[19] P. Kumar and S. L. Ta, "Face Recognition Attendance System Using Local Binary Pattern Algorithm," in *2nd International Conference on Vision Towards Emerging Trends in Communication and Networking Technologies*, Vellore, India, Dec. 2023, pp. 1–6, https://doi.org/10.1109/ViTECoN58111.2023.10157843.

[20] A. B. S. Salamh and H. I. Akyüz, "A Novel Feature Extraction Descriptor for Face Recognition," *Engineering, Technology & Applied*

Science Research, vol. 12, no. 1, pp. 8033–8038, Feb. 2022, https://doi.org/10.48084/etasr.4624.

[21] S. Naseem, S. S. Rathore, S. Kumar, S. Gangopadhyay, and A. Jain, "An approach to occluded face recognition based on dynamic image-to-class warping using structural similarity index," *Applied Intelligence*, vol. 53, no. 23, pp. 28501–28519, Dec. 2023, https://doi.org/10.1007/s10489-023-05026-0.

[22] Z. Song, X. Yang, Z. Xu, and I. King, "Graph-Based Semi-Supervised Learning: A Comprehensive Review," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 34, no. 11, pp. 8174–8194, Aug. 2023, https://doi.org/10.1109/TNNLS.2022.3155478.

[23] J. Wang, Y. Guo, X. Wen, Z. Wang, Z. Li, and M. Tang, "Improving graph-based label propagation algorithm with group partition for fraud detection," *Applied Intelligence*, vol. 50, no. 10, pp. 3291–3300, Oct. 2020, https://doi.org/10.1007/s10489-020-01724-1.

[24] C. Zhang, T. Bai, and B. Wu, "Semi-supervised Graph Learning with Few Labeled Nodes," in *International Conference on Database Systems for Advanced Applications*, Apr. 2022, pp. 423–438, https://doi.org/10.1007/978-3-031-00126-0_32.

[25] N. Reyaz, G. Ahamad, N. J. Khan, M. Naseem, and J. Ali, "SVMCTI: support vector machine based cricket talent ıdentification model," *International Journal of Information Technology*, vol. 16, no. 3, pp. 1931–1944, Mar. 2024, https://doi.org/10.1007/s41870-023-01686-w.

[26] J. Wang and Y. Dong, "Measurement of Text Similarity: A Survey," *Information*, vol. 11, no. 9, Sep. 2020, Art. no. 421, https://doi.org/10.3390/info11090421.

[27] A. D. Sokolova and A. V. Savchenko, "Computation-Efficient Face Recognition Algorithm Using a Sequential Analysis of High Dimensional Neural-Net Features," *Optical Memory and Neural Networks*, vol. 29, no. 1, pp. 19–29, Jan. 2020, https://doi.org/10.3103/S1060992X2001004X.

[28] D. E. King, "Dlib-ml: A Machine Learning Toolkit," *The Journal of Machine Learning Research*, vol. 10, pp. 1755–1758, Sep. 2009.

[29] W. S. Ow, M. A. Ilyas, N. H. Kamarudin, M. B. Othman, Z. B. Zulkoffli, and Y. B. Chu, "Face Recognition Authentication System with CNN and Blink Detection Algorithm," in *International Conference on Computing, Control and Industrial Engineering*, Hangzhou, China, Feb. 2023, pp. 491–501, https://doi.org/10.1007/978-981-99-2730-2_48.

[30] "make_circles," *scikit-learn*. https://scikit-learn/stable/modules/generated/sklearn.datasets.make_circles.html.

[31] "The Database of Faces." 2001, [Online]. Available: https://cam-orl.co.uk/facedatabase.html.

[32] "Yale Face Database." 1997, [Online]. Available: http://cvc.cs.yale.edu/cvc/projects/yalefaces/yalefaces.html.

[33] "color FERET Database." NIST, Jan. 31, 2011, [Online]. Available: https://www.nist.gov/itl/products-and-services/color-feret-database.