Clustering Optimization via Centroid Neural Network Ensemble

Chung Nguyen Tran
Departament de MiSE
Universitat Autònoma de Barcelona
Bellaterra, Spain
chungnguyen.tran@autonoma.cat

Le-Anh Tran*
Dept. of Research and Development
Mindintech, Inc.
Seoul, South Korea
latran@mindintech.com

Ngoc-Phu Le
Dept. of Data and Analytics
Unilever PLC
Ho Chi Minh City, Vietnam
lephu0803@gmail.com

Jordi Carrabina and David Castells-Rufas
Departament de MiSE
Universitat Autònoma de Barcelona
Bellaterra, Spain
jordi.carrabina@uab.cat, david.castells@uab.cat

Minh Son Nguyen
Dept. of Computer Engineering
University of Information Technology
Ho Chi Minh City, Vietnam
sonnm@uit.edu.vn

Nhan Cach Dang*
IITEEE
HCMC University of Transport
Ho Chi Minh City, Vietnam
cach@ut.edu.vn

***Abstract*—A cluster analysis approach based on an ensemble of unsupervised competitive neural networks for clustering optimization is proposed in this paper. The proposed method utilizes centroid neural network (CentNN) as its backbone, which has been recognized as an effective data clustering algorithm that can yield more satisfactory clustering error rates compared to conventional methods. In an attempt to further enhance the clustering results, this study investigates a centroid neural network ensemble (CentNN-E). By leveraging the strengths of multiple CentNN models, the proposed CentNN-E scheme aims to mitigate individual biases and achieve more robust and reliable clustering results. For one specific cluster, the prototypes (centroid candidates) produced by multiple CentNN models (with different initializations) are grouped by using Euclidean distance and are convexly combined using error-based weighting on a per-cluster basis. The resulting prototypes from the ensemble are subsequently utilized as initial prototypes for a final execution. Experimental results on various synthetic test data sets demonstrate that the proposed CentNN-E approach can yield superior results and surpass a single CentNN algorithm in terms of clustering error criterion.**

***Index Terms*—Cluster analysis, clustering, unsupervised learning, centroid neural network, model ensemble.**

I. Introduction

Data clustering is an unsupervised learning method used in data analysis to automatically categorize and organize a set of individual data points based on their shared characteristics. Unlike certain types of machine learning techniques, clustering does not require data points to have predefined labels. Instead, it identifies patterns and structures within the data set itself, making it particularly useful for exploring hidden patterns in the data. There exist various genres of clustering algorithms designed for different objectives. The most fundamental type of clustering is partitional methods, which prioritize minimizing the clustering error (the sum of Euclidean distances from all cluster centers to their respective data points). Notable partitional clustering approaches including K-Means [1], K-Means++ [2], Fuzzy C-Means (FCM) [3], POCS-based [4], [5], and Centroid Neural Network (CentNN) [6] algorithms have been commonly applied in numerous applications [7]–[14]. Among these approaches, the CentNN method, an unsupervised learning algorithm inspired by the well-known K-Means clustering method, has been acknowledged as an effective approach that is capable of reducing clustering error by applying competitive learning with the concepts of winner/loser neurons. The advantages of CentNN include effectiveness and stability, as it provides a certain level of consistency in clustering outcomes, thereby diminishing sensitivity to the initial placement of cluster centroids. However, similar to any unsupervised algorithms, the CentNN approach also has its own drawback: the potential to converge to a local minimum instead of an optimal clustering solution for the given data.

Inspired by the observations made on the CentNN algorithm, in this paper, we propose a centroid neural network ensemble model, called CentNN-E, to enhance clustering outcomes. The proposed CentNN-E scheme combines the strengths of multiple CentNN models in order to mitigate individual biases and achieve more robust and dependable results. For one specific cluster, the prototype candidates generated by various CentNN models (with different initial prototypes) are grouped based on Euclidean distance and then are convexly combined using error-based weighting on a per-cluster basis. These ensemble prototypes are subsequently used as initial prototypes for a final execution. Through the integration of insights from multiple models, the ensemble aims to improve overall accuracy and minimize errors inherent in individual models. Experimental results on various synthetic data sets demonstrate that the proposed CentNN-E scheme can outperform the CentNN algorithm and produce improved results in terms of the total clustering error.

The remainder of this paper is organized as follows: Section

*Corresponding authors.

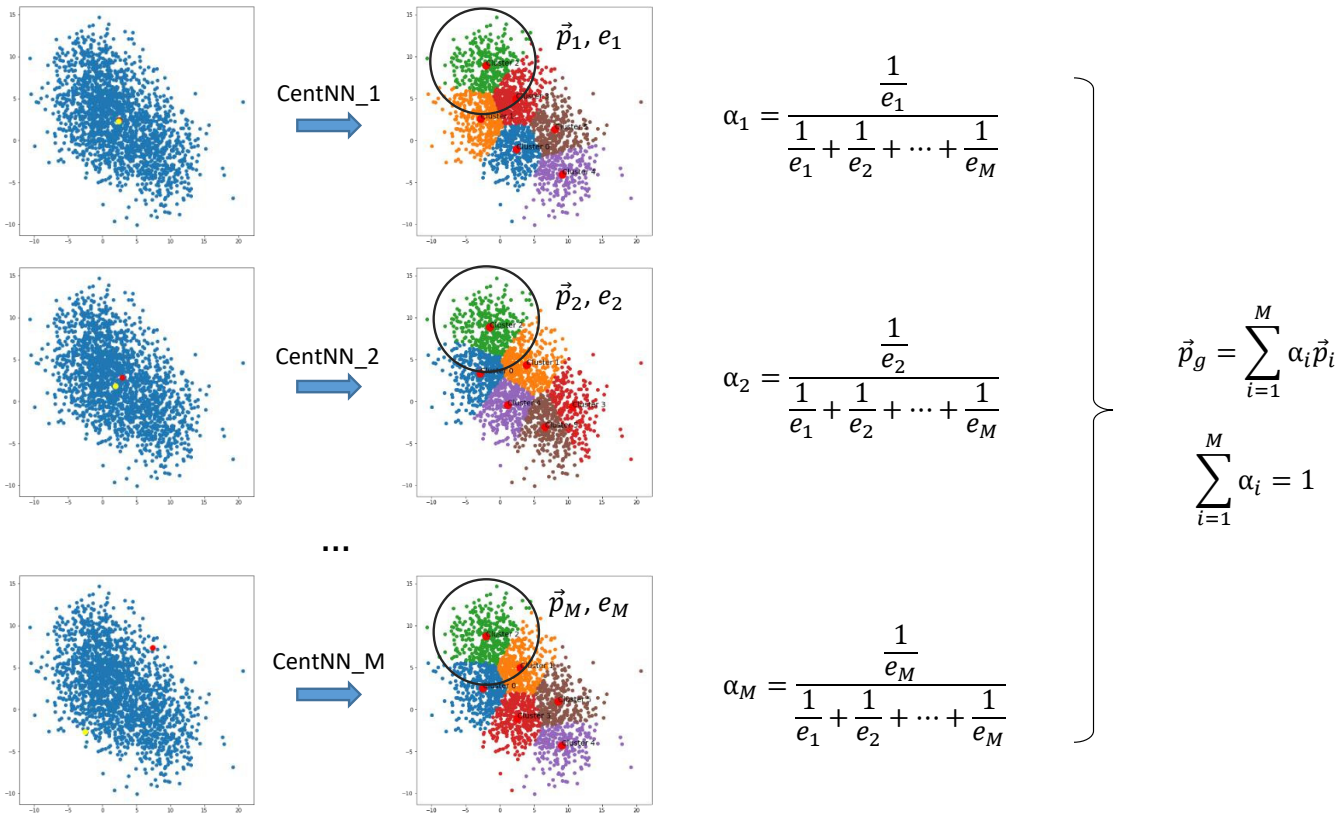


Fig. 1: The prototype candidates for one cluster can be grouped based on Euclidean distance and then are convexly combined using error-based weighting on a per-cluster basis.

II briefly describes the preliminaries of this research. The proposed CentNN-E scheme is elaborated in Section III. The results and analyses on various synthetic data sets are presented in Section IV. Section V concludes the paper.

II. Preliminaries

In this section, we briefly review the preliminaries that influence this study including the CentNN algorithm and the model ensemble method.

A. Centroid Neural Network (CentNN)

The CentNN algorithm [6] is a clustering approach that identifies data prototypes within corresponding clusters for all data vectors presented. Instead of recalculating the prototypes of all clusters for every data presentation, the CentNN method updates its weights only when the status of the considered neuron in the current epoch differs from that in the previous epoch. Specifically, when an input vector data $\vec{x}$ is introduced into the network at epoch q, the winner neuron is determined as the neuron with the minimum distance to $\vec{x}$. Conversely, the loser neuron is defined as the neuron that was the winner neuron for $\vec{x}$ at epoch $q-1$ but is not the winner neuron for $\vec{x}$ at the current epoch q. The objective function of the CentNN algorithm can be expressed as:

$$E = \sum_{k=1}^{C} E_k = \sum_{k=1}^{C} \sum_{n=1}^{N_k} ||\vec{x}_{kn} - \vec{p}_k||, \tag{1}$$

where C, N_k, and $\vec{p}_k$ denote the number of clusters, the number of data points in the cluster k, and the prototype of the cluster k, respectively. The final set of prototypes, $\{\vec{p}_k, 1 \leq k \leq C\}$, is obtained by minimizing the objective function Eq. (1). The adaptive equations for updating winner neuron w and loser neuron l can be written as follows:

$$\vec{p}_w[q+1] = \vec{p}_w[q] + \frac{1}{N_w+1}(\vec{x} - \vec{p}_w[q]), \tag{2}$$

$$\vec{p}_l[q+1] = \vec{p}_l[q] - \frac{1}{N_l-1}(\vec{x} - \vec{p}_l[q]), \tag{3}$$

where $\vec{p}_w$ and $\vec{p}_l$ denote the prototypes of the winner neuron and the loser neuron, respectively. Note that the CentNN algorithm only updates prototypes when an input vector changes its cluster membership [6].

B. Ensemble Learning

In the field of machine learning, model ensemble (or ensemble learning) [15] is a powerful technique that combines

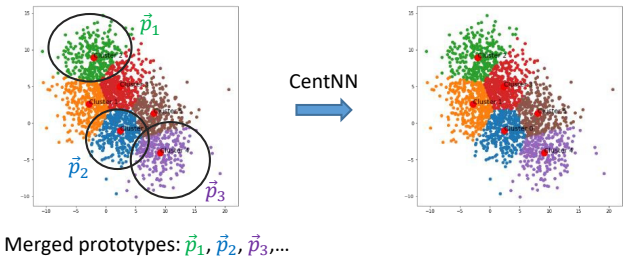


Fig. 2: The merged prototype candidates are utilized as initial prototypes for one last execution.

predictions from multiple models to generate a more generalizable and reliable outcome. Model ensemble provides a variety of perspectives and leverages the collective information of multiple models to improve performance, as individual models might have their strengths and weaknesses. For instance, if certain models are overly sensitive to specific variations in the data, ensemble learning can help average out these sensitivities, resulting in a more stable prediction.

Ensemble learning offers several advantages. Firstly, by combining the predictions of multiple models, ensembles often achieve higher accuracy compared to any single model on its own. Additionally, ensemble methods can effectively address the issue of overfitting, which occurs when a model performs well on the training data but poorly on unseen data. Furthermore, ensembles are less susceptible to errors caused by individual models, making them more robust when dealing with outliers and noise in the data. Ensemble learning is particularly beneficial for improving the performance of machine learning models in tasks such as classification and regression. However, it is crucial to consider the trade-off between accuracy and complexity, since ensembles can be more computationally expensive to train and interpret compared to single models.

III. Methodology

In this section, we elaborate on the pipeline of the proposed ensemble scheme. Given that the constraints of execution time are not a concern, the objective is to effectively optimize the clustering error. The proposed scheme utilizes the CentNN algorithm as the backbone and adopts an ensemble method to achieve improved overall accuracy and mitigate the errors and biases present in any individual model.

Given two integers M and C representing the number of CentNN members and the predetermined number of clusters, respectively, the proposed scheme consists of several key steps which are described as follows. Initially, multiple CentNN members independently perform conventional clustering with different initial prototypes on the given data set, resulting in M sets, with each set containing C prototypes generated by each member after convergence. Subsequently, prototype grouping is carried out which involves identifying C groups of prototypes. Each group has M prototypes which are yielded by M separate CentNN members such that these M prototypes

Algorithm 1 The CentNN-E Scheme.

Input: Dataset, number of clusters C, number of CentNN members M
Output: Cluster prototypes
Steps:
for $i = 1, ..., M$ **do**
  Apply the i^{th} CentNN member to the dataset
  Append prototype candidates to a list
  Append respective clustering errors to a list
end
Merge prototypes
Apply CentNN using the set of merged prototypes as initial prototypes

TABLE I: Descriptions of the synthetic data sets used for experiments.

Datasets	#Instances	#Attributes	#Clusters
Face Embeddings	118	128	5
Lena Image Blocks	4.096	64	8
MNIST	10,000	784	10

are closest to one another in terms of Euclidean distance compared to the remaining prototypes. These M prototypes are then convexly combined using error-based weighting on a per-cluster basis, as illustrated in Fig. 1. That is, given that $\{\vec{p}_i, 1 \leq i \leq M\}$ represents one set of prototypes after grouping, and $\{e_i, 1 \leq i \leq M\}$ denotes a set of corresponding clustering errors. The merged prototype of one group can be calculated as:

$$\vec{p}_g = \sum_{i=1}^{M} \alpha_i \vec{p}_i, \tag{4}$$

where α_i represents the weight for $\vec{p}_i$, which is calculated as:

$$\alpha_i = \frac{\frac{1}{e_i}}{\sum_{j=1}^{M} \frac{1}{e_j}}. \tag{5}$$

The purpose of the weight calculating equation stated in Eq. (5) is to prioritize the contribution of prototypes with lower error while minimizing the influence of prototypes with higher error in the calculation of the merged prototype. In the final step, the proposed scheme utilizes the obtained set of merged prototypes as initial prototypes and runs the CentNN algorithm one more time to determine the final set of cluster prototypes, as shown in Fig. 2. The pseudocode of the proposed scheme is shown in Algorithm 1.

IV. Experiments

In this section, experiments and analyses are conducted to evaluate the effectiveness of the proposed method. The synthetic data sets used for experiments are first described, then the performance of the proposed scheme is discussed.

TABLE II: Clustering error on Face Embeddings for 10 independent runs.

	Member 1	Member 2	Member 3	Member 4	Member 5	CentNN-E
1st	172.43	172.43	172.43	172.43	172.43	172.43
2nd	172.43	172.43	172.43	172.43	172.43	172.43
3rd	172.43	172.43	172.43	231.82	172.43	172.43
4th	172.43	172.43	172.43	172.43	231.82	172.43
5th	172.43	172.43	172.43	172.43	172.43	172.43
6th	172.43	230.56	172.43	172.43	172.43	172.43
7th	172.43	172.43	230.83	172.43	172.43	172.43
8th	172.43	172.43	230.62	172.43	172.43	172.43
9th	172.43	172.43	172.43	172.43	172.43	172.43
10th	172.43	172.43	231.01	172.43	172.43	172.43

TABLE III: Clustering error on Lena Image Blocks for 10 independent runs.

	Member 1	Member 2	Member 3	Member 4	Member 5	CentNN-E
1st	88.42	88.42	91.33	88.48	89.07	90.05
2nd	90.04	93.46	91.86	91.86	90.05	90.00
3rd	89.13	93.53	93.53	88.56	93.19	88.99
4th	88.45	91.71	88.52	88.43	88.48	88.23
5th	88.38	91.58	97.46	88.42	91.19	89.62
6th	88.11	92.26	92.81	92.66	92.74	88.17
7th	88.48	96.03	88.49	88.49	91.90	88.42
8th	89.99	94.44	91.85	91.77	94.45	89.73
9th	88.25	89.97	92.79	89.80	89.98	89.91
10th	88.48	88.38	93.54	93.60	92.81	88.33

TABLE IV: Clustering error on MNIST dataset for 10 independent runs.

	Member 1	Member 2	Member 3	Member 4	Member 5	CentNN-E
1st	1976.88	2008.44	2007.64	2111.23	2007.86	1935.42
2nd	1976.32	1975.92	2009.48	1975.57	1976.84	1943.21
3rd	1972.42	1972.47	1976.87	1972.45	1975.99	1965.51
4th	1976.36	1976.87	2007.76	1997.16	1976.72	1949.59
5th	1977.16	2129.16	1976.67	2009.51	1975.74	1968.11
6th	1972.42	1971.99	1972.81	2110.57	1972.96	1968.44
7th	1949.44	2010.26	2013.46	2001.74	2114.88	1968.99
8th	1975.92	1977.14	2007.64	2037.22	1975.81	1972.17
9th	1970.77	2069.04	2190.09	1973.88	1977.15	1972.44
10th	1976.71	1976.71	1976.64	2036.98	2014.52	1944.12

A. Data Preparation

In order to evaluate the performance of the proposed scheme, experiments have been conducted on various synthetic data sets, whose specifications are summarized in Table I.

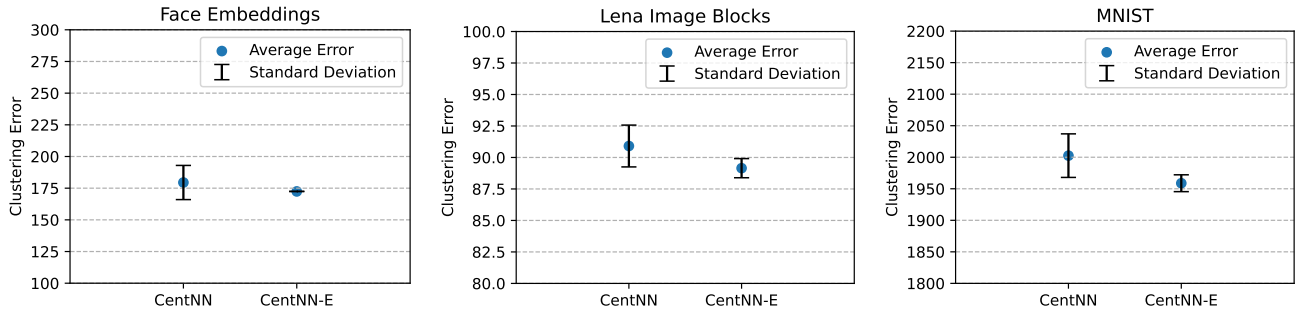
Face Embeddings: The first dataset utilized in this research is the Five Celebrity Faces dataset [16], a small dataset comprising images of five celebrities: Ben Afflek, Elton John, Jerry Seinfeld, Madonna, and Mindy Kaling. This dataset has two sets of face images for training and validation. However, due to its small size, the two image sets are combined to create a unified dataset of 118 images for clustering purposes. A pre-trained FaceNet model [17] is applied to generate 118 corresponding feature embeddings for all the images. FaceNet takes as input a 160×160 RGB image and generates a 128×1 embedding. Accordingly, a collection of 118 facial embeddings with a dimension of 128×1 is derived and used for clustering experiments.

Lena Image Blocks: The second data set is synthesized from the well-known Lena image with a resolution of 512×512 pixels. Specifically, The image is partitioned into blocks with a block size of 8×8, these blocks are then flattened to generate a set of 64×1 embeddings. Accordingly, a total of 4,096 data vectors is obtained.

MNIST: The MNIST dataset [18] comprises 60,000 and 10,000 gray-scale images for training and validation, respectively, with each image having a resolution of 28×28 pixels. For the clustering experiments, the validation set is utilized and all the validation images are reshaped into 784×1 to be used as input data.

B. Results and Analyses

We first analyze the proposed CentNN-E scheme and its CentNN members in terms of clustering error, as stated in Eq. (1). The number of CentNN members is set to 5 and the experiment on each data set is executed 10 times with the numerical results reported in Table II, Table III, and Table IV. In the case of the Face Embeddings data set, Table II reveals that the proposed CentNN-E scheme consistently achieves stable convergence results with similar clustering error rates across all 10 executions, while certain CentNN members occasionally



(a) (b) (c)

Fig. 3: Comparisons between CentNN and CentNN-E in terms of average and standard deviation of clustering error.

produce unstable and elevated error rates. The results for Lena Image data, as presented in Table III, indicate that the error rates given by the proposed CentNN-E scheme are frequently lower than those yielded by its CentNN members. This trend can also be witnessed in the outcomes for the MNIST dataset, as summarized in Table IV. It is clear that the proposed CentNN-E scheme generally surpasses its CentNN members in this experiment by consistently yielding the lowest clustering error in nearly all scenarios.

In order to provide a more comprehensive comparison, a thorough analysis comparing the proposed CentNN-E scheme with a single CentNN algorithm is conducted. Specifically, each scheme is executed separately 20 times on the three aforementioned data sets. The results, including the average and standard deviation values of clustering error, are presented in Fig. 3, showcasing that the proposed CentNN-E scheme outperforms the single CentNN method in all three cases, presenting significant improvements in both average and standard deviation values of clustering error. Notably, on the Face Embeddings data, the proposed CentNN-E scheme achieves zero deviation after 20 executions. This implies that the proposed CentNN-E scheme can produce a stable convergence process when compared to a single CentNN approach.

V. Conclusions

In this paper, a centroid neural network (CentNN)-based ensemble scheme for clustering optimization is proposed. The CentNN algorithm is a clustering approach that is acknowledged for its prowess in demonstrating superior clustering error rates compared to other conventional methods. To further enhance clustering results, this study explores a centroid neural network ensemble (CentNN-E) to capitalize on the strengths of multiple CentNN models so as to mitigate biases present in any individual model and achieve more accurate results. The prototype candidates generated by various CentNN models for a specific cluster are grouped based on Euclidean distance and combined convexly using error-based weighting on a per-cluster basis. The merged prototypes are afterward adopted as initial prototypes for a final execution. Empirical results on various synthetic test data sets have shown that the proposed CentNN-E approach surpasses the CentNN algorithm and produces superior clustering error results.

Acknowledgment

This research has been partly funded by Catalan Government Research Groups ref. 2021-SGR-01623.

References

[1] J. A. Hartigan and M. A. Wong, "Algorithm as 136: A k-means clustering algorithm," *Journal of the royal statistical society. series c (applied statistics)*, vol. 28, no. 1, pp. 100–108, 1979. 1

[2] D. Arthur and S. Vassilvitskii, "K-means++ the advantages of careful seeding," in *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*, 2007, pp. 1027–1035. 1

[3] J. C. Bezdek, R. Ehrlich, and W. Full, "Fcm: The fuzzy c-means clustering algorithm," *Computers & geosciences*, vol. 10, no. 2-3, pp. 191–203, 1984. 1

[4] L.-A. Tran, H. M. Deberneh, T.-D. Do, T.-D. Nguyen, M.-H. Le, and D.-C. Park, "Pocs-based clustering algorithm," in *2022 International Workshop on Intelligent Systems (IWIS)*. IEEE, 2022, pp. 1–6. 1

[5] L.-A. Tran, D. Kwon, H. M. Deberneh, and D.-C. Park, "Cluster analysis via projection onto convex sets," *Intelligent Data Analysis*, no. Preprint, pp. 1–18, 2024. 1

[6] D.-C. Park, "Centroid neural network for unsupervised competitive learning," *IEEE Transactions on Neural Networks*, vol. 11, no. 2, pp. 520–528, 2000. 1, 2

[7] D.-C. Park and Y.-J. Woo, "Weighted centroid neural network for edge preserving image compression," *IEEE transactions on neural networks*, vol. 12, no. 5, pp. 1134–1146, 2001. 1

[8] M. T. Ngoc and D.-C. Park, "Centroid neural network with pairwise constraints for semi-supervised learning," *Neural Processing Letters*, vol. 48, no. 3, pp. 1721–1747, 2018. 1

[9] L.-A. Tran and M.-H. Le, "Robust u-net-based road lane markings detection for autonomous driving," in *2019 International Conference on System Science and Engineering (ICSSE)*. IEEE, 2019, pp. 62–66. 1

[10] L.-A. Tran, T.-D. Nguyen, T.-D. Do, C. N. Tran, D. Kwon, and D.-C. Park, "Embedding clustering via autoencoder and projection onto convex set," in *2023 International Conference on System Science and Engineering (ICSSE)*. IEEE, 2023, pp. 128–133. 1

[11] Y.-S. Song, D.-C. Park, C. N. Tran, H.-S. Choi, and M. Suk, "Fuzzy c-means algorithm with divergence-based kernel," in *International Conference on Fuzzy Systems and Knowledge Discovery*. Springer, 2006, pp. 99–108. 1

[12] D.-C. Park, C. N. Tran, and S. Park, "Gradient based fuzzy c-means algorithm with a mercer kernel," in *International Symposium on Neural Networks*. Springer, 2006, pp. 1038–1043. 1

[13] D.-C. Park, C. N. Tran, and Y. Lee, "Content-based classification of images using centroid neural network with divergence measure," in *Australasian Joint Conference on Artificial Intelligence*. Springer, 2006, pp. 729–738. 1

[14] T.-D. Do, L.-A. Tran, T.-D. Nguyen, N.-N. Truong, D.-C. Park, and M.-H. Le, "Pocs-based image compression: An empirical examination," in *2024 7th International Conference on Green Technology and Sustainable Development (GTSD)*. IEEE, 2024. 1

[15] D. Opitz and R. Maclin, "Popular ensemble methods: An empirical study," *Journal of artificial intelligence research*, vol. 11, pp. 169–198, 1999. 2

[16] "Five celebrity faces dataset," https://www.kaggle.com/datasets/dansbecker/5-celebrity-faces-dataset. 4

[17] F. Schroff, D. Kalenichenko, and J. Philbin, "Facenet: A unified embedding for face recognition and clustering," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 815–823. 4

[18] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998. 4