Embedding Clustering via Autoencoder and Projection onto Convex Set

Le-Anh Tran
Dept. of Electronics Engineering
Myongji University
Gyeonggi, South Korea
leanhtran@mju.ac.kr

Thanh-Dat Nguyen
Dept. of Research and Development
OCST Co., Ltd.
Seoul, South Korea
thanhdat6716@gmail.com

Truong-Dong Do
Dept. of Aerospace Systems Engineering
Sejong University
Seoul, South Korea
dongdo@sju.ac.kr

Chung Nguyen Tran
Dept. of Research and Development
IriTech, Inc.
Ho Chi Minh City, Vietnam
tnchung@iritech.com

Daehyun Kwon
Dept. of ITPM
Soongsil University
Seoul, South Korea
isifter@gmail.com

Dong-Chul Park*
Dept. of Electronics Engineering
Myongji University
Gyeonggi, South Korea
parkd@mju.ac.kr

***Abstract*—Projection onto Convex Set (POCS) is a powerful signal processing tool for various convex optimization problems. For non-intersecting convex sets, the simultaneous POCS method can result in a minimum mean square error solution. This property of POCS has been applied to clustering analysis and the POCS-based clustering algorithm was proposed earlier. In the POCS-based clustering algorithm, each data point is treated as a convex set, and a parallel projection operation from every cluster prototype to its corresponding data members is carried out in order to minimize the objective function and to update the memberships and prototypes. The algorithm works competitively against conventional clustering methods in terms of execution speed and clustering error on general clustering tasks. In this paper, the performance of the POCS-based clustering algorithm on a more complex task, embedding clustering, is investigated in order to further demonstrate its potential in benefiting other high-level tasks. To this end, an off-the-shelf FaceNet model and an autoencoder network are adopted to synthesize two sets of feature embeddings from the Five Celebrity Faces and MNIST datasets, respectively, for experiments and analyses. The empirical evaluations show that the POCS-based clustering algorithm can yield favorable results when compared with other prevailing clustering schemes such as the K-Means and Fuzzy C-Means algorithms in embedding clustering problems.**

***Index Terms*—POCS-based clustering, machine learning, unsupervised learning, high-dimensional data, MNIST**

I. Introduction

Clustering is an unsupervised data analysis technique that aims to segregate data points in a given dataset that have similar traits and assign them to clusters [1]. Clustering is one of the most fundamental tasks that has been widely deployed in numerous machine learning-driven automation systems [2]–[4]. Popular clustering approaches find homogeneous subgroups of data points that have similar characteristics based on optimizing some predefined criteria. The most well-known criterion is the clustering error measure which is defined as the sum of the distances from cluster centers to all of their corresponding member points [5]. In this sense, one of the most classical methods for general clustering tasks is the K-Means clustering algorithm, which applies the Euclidean distance to measure the similarities among data points [1]. The K-Means algorithm alternates between assigning cluster membership for each data point to the nearest cluster center and computing the center of each cluster as the prototype of its member data points. The K-Means algorithm's training procedure is terminated when there is no further update in the assignment of instances to clusters. However, the convergence process of the K-Means algorithm is considerably dependent on the initialized prototypes, in addition, this classical clustering algorithm is known to be sensitive to noise and outliers [5]–[7].

Another widely used clustering algorithm is the Fuzzy C-Means (FCM) clustering algorithm [8]. Unlike the K-Means algorithm, a data point can concurrently belong to multiple subgroups in the FCM algorithm. A membership function is adopted to represent the degree of certainty for whether a data point belongs to a certain cluster. The performance of the FCM algorithm, however, is also highly dependent on the selection of the initial prototypes and membership values [8]. Moreover, the disadvantages of the FCM clustering algorithm include extended computational time and incapability in handling noisy data and outliers [8]. In order to handle the computational complexity and upgrade the convergence speed of the FCM algorithm, Park and Dagher introduced the Gradient-Based Fuzzy C-Means (GBFCM) algorithm [9] in which the minimization process of the objective function is proceeded by solving two equations alternatively in an iterative fashion.

On the other hand, *Projection onto Convex Set* (POCS) is a robust tool for signal synthesis and image restoration which was introduced by Bregman in the mid-1960s [10]. Bregman

*Corresponding Author

979-8-3503-2294-1/23/$31.00 ©2023 IEEE

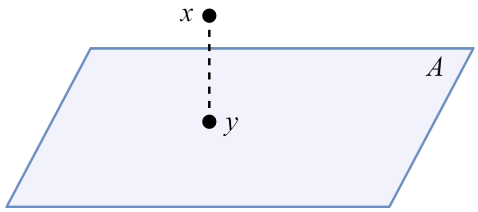

Fig. 1. The projection of a point x onto a set A ($x \notin A$) is the unique element y in A which is closest to x.

has shown that successive projections between two or more intersecting convex sets converge to a point that exists in the intersection of the convex sets. If the convex sets are disjoint, the sequential projections converge to greedy limit cycles which are dependent on the order of the projections. For non-intersecting convex sets, the method of simultaneous projections can result in a minimum mean square error solution [11]. This convergence property of the POCS method has been applied to clustering problems and the POCS-based clustering algorithm was proposed in [12] which has been proved to be able to perform competitively when compared with other conventional clustering approaches such as the K-Means and FCM algorithms. The POCS-based clustering algorithm treats each data point as a convex set and projects the prototype of every cluster to each of its constituent instances. The projections are convexly combined to optimize and compute a new set of center prototypes and to minimize the objective function. In this paper, we further examine the effectiveness and efficiency of the POCS-based clustering algorithm for complex clustering tasks such as feature embedding clustering in order to demonstrate its potential in benefiting other high-level tasks [13]. To this end, two sets of feature embeddings are prepared for experiments and analyses. The first set is obtained by utilizing an off-the-shelf FaceNet model [14] to extract embeddings from the image data of the Five Celebrity Faces dataset [15], and the other set is prepared by training an autoencoder network to learn the representations of handwritten digit images of the MNIST database [16].

The rest of this paper is structured as follows. Section II briefly reviews the concepts of convex sets and the POCS-based clustering algorithm. The process of preparing feature embedding datasets is presented in Section III. In Section IV, the performance of the POCS-based clustering algorithm on feature embedding clustering tasks is examined and compared with those of other prevailing clustering approaches including the K-Means and FCM algorithms. Section V concludes the paper.

II. Preliminaries

A. Convex Set

Convex set has been one of the most classical and powerful concepts in optimization theory [10]. A set of data points is called a convex set if it has the following property: given a non-empty set A which is the subset of a Hilbert space H ($A \subseteq H$), $\forall x_1, x_2 \in A$ and $\forall \lambda \in [0, 1]$, A is convex if the following holds true:

$$x := \lambda x_1 + (1 - \lambda) x_2 \in A. \tag{1}$$

Note that if $\lambda = 1$, then $x = x_1$, and if $\lambda = 0$, then $x = x_2$. In this sense, x lies on the line segment joining x_1 and x_2 when the set is convex.

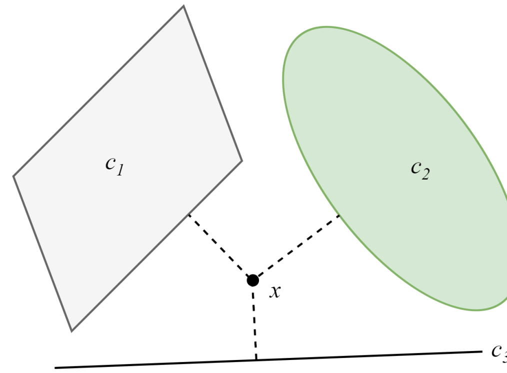

Fig. 2. Graphical interpretation of parallel POCS for disjoint convex sets.

B. Projection onto Convex Set (POCS)

The concept of projecting a point to a plane is utilized to solve many optimization problems such as finding a point on the plane that has the minimum distance from the center of projection. For a given point $x \notin A$, the projection of x onto A is the unique point $y \in A$ such that the distance between x and y is minimum. If $x \in A$, the projection of x onto A is x. The optimization task can be expressed as:

$$y = argmin||x - y^*|| \tag{2}$$

where y^* denotes all the points that belong to the set A. A graphical illustration of the projection of a point onto a convex set is depicted in Fig. 1.

C. Parallel POCS

In the parallel POCS method, a point is projected to all convex sets concurrently. All the projections are combined convexly with corresponding weight values to solve minimization problems. Given a set of n convex sets $C = \{c_i | 1 \leq i \leq n\}$, the convergence of the simultaneous weighted projections can be computed as follows:

$$x_{p+1} = x_p + \sum_{i=1}^{n} w_i (P_{c_i} - x_p), \quad p = 0, 1, 2, ... \tag{3}$$

where x_p represents the p^{th} projection from the initial point x_0, P_{c_i} is the projection of x_p onto convex set c_i and w_i is the weight of importance of the projection such that:

$$\sum_{i=1}^{n} w_i = 1. \tag{4}$$

Fig. 3. Face image samples of one class from the Five Celebrity Faces dataset.

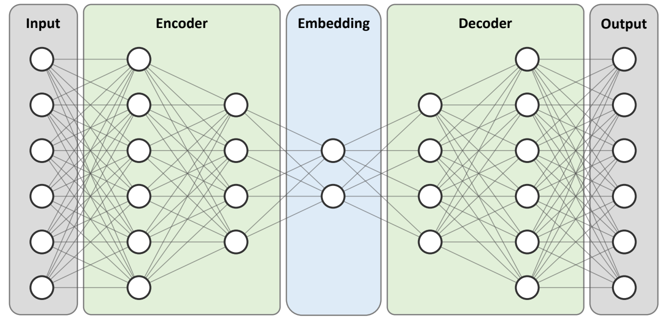

Fig. 4. Typical diagram of an autoencoder network.

The main advantages of the parallel mode of POCS when compared with the alternating one include computational efficiency and improved execution time [12]. When the convex sets are disjoint, the parallel POCS method converges to a point that minimizes the weighted sum of the distances from the point to the sets, which can be expressed as:

$$x_\infty = argmin \sum_{i=1}^{n} w_i ||x - P_{c_i}|| \tag{5}$$

where x_∞ is the convergence point. A graphical illustration of the parallel POCS method is presented in Fig. 2.

D. POCS-based Clustering Algorithm

For disjoint convex sets, the parallel mode of the POCS method converges to a minimum mean square error solution [11]. This property has been applied to clustering problems and the POCS-based clustering algorithm has been proposed earlier in [12]. The POCS-based clustering algorithm considers each data point as a convex set and all data points in the cluster as disjoint convex sets. Given a set of data points with a predefined number of clusters k, the objective function of the algorithm is defined as:

$$J = \sum_{j}^{k} \sum_{i=1}^{n_j} w_{ji} ||x_j - P_{ji}|| \tag{6}$$

in which the importance weight w_{ji} is computed as:

$$w_{ji} = \frac{||x_j - d_{ji}||}{\sum_{m=1}^{n_j} ||x_j - d_m||} \tag{7}$$

where n_j denotes the number of data points in one cluster, while P_{ji} is the projection of the cluster prototype x_j onto the member point d_{ji}.

At the beginning, the POCS-based clustering algorithm initializes k cluster prototypes by adopting the prototype initialization method of the K-Means++ algorithm [17], then based on the Euclidean distance to the prototypes, each data point is assigned to one of the clusters which has the minimum distance to the data point. Until convergence, the algorithm computes new cluster prototypes using the following equation:

$$x_{j,p+1} = x_{j,p} + \sum_{i=1}^{n_j} w_{ji,p}(P_{ji,p} - x_{j,p}), \quad p = 0, 1, 2, ... \tag{8}$$

where p is the iteration index. Starting from an initial point $x_{j,0}$, the projections converge to a point, $x_{j,\infty}$, that can minimize the objective function (6).

III. Data Preparation

This section presents the preparation of feature embedding data. In order to evaluate the robustness and applicability of the POCS-based clustering algorithm to feature embedding clustering tasks, experiments on two synthetic feature embedding datasets have been conducted. We adopt FaceNet [14] and a plain autoencoder model [18] to extract feature embeddings from the Five Celebrity Faces [15] and MNIST [16] datasets for experiments and analyses.

A. FaceNet

FaceNet [14] is a face recognition model proposed by researchers at Google in 2015 that has achieved state-of-the-art results on various face recognition benchmarks. FaceNet can be utilized to extract high-quality face embeddings from input facial images, those embeddings afterward can be used to train and develop a face recognition system.

The first dataset used in this study is the Five Celebrity Faces dataset [15], which is a small dataset containing the photos of five celebrities: Ben Afflek, Elton John, Jerry Seinfeld, Madonna, and Mindy Kaling. The dataset is divided into training and validation sets. However, as this is a small dataset, we merge the two image sets to obtain a single dataset of 118 images for clustering tasks. FaceNet is adopted to extract feature embeddings from all 118 face images. The input and output shapes of FaceNet are 160x160x3 and 128x1, respectively. As the result, we can obtain 118 face embeddings with a size of 128x1 which is used as the input data for the clustering experiments. Several image samples of one class from the Five Celebrity Faces dataset are shown in Fig. 3.

B. Autoencoder

Autoencoder (AE) [18] is a type of artificial neural network that is used to learn efficient embeddings of unlabeled data. A general AE model is comprised of three main components: encoder, code, and decoder, as shown in Fig. 4. The input and output of an AE network generally have the same shape. The encoder compresses input data into a low-dimensional code (or representation) and the decoder reconstructs the code to

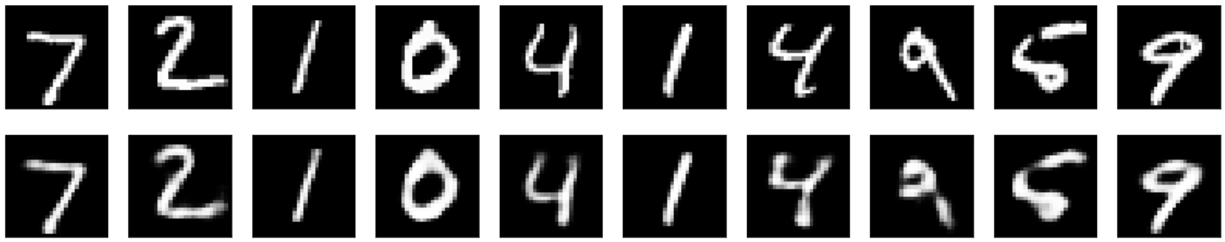

Fig. 5. Reconstructed images on MNIST dataset using the autoencoder model described in the paper (top: input image, bottom: reconstructed image).

TABLE I
DESCRIPTION OF THE AE MODEL USED IN THIS STUDY.

Part	Layer	Input Shape	Output Shape	#Params
Input	Input	784x1	784x1	0
Encoder	Dense	784x1	128x1	100,480
	Dense	128x1	64x1	8,256
	Dense	64x1	32x1	2,080
Embedding		32x1	32x1	0
Decoder	Dense	32x1	64x1	2,112
	Dense	64x1	128x1	8,320
	Dense	128x1	784x1	101,136
Output	Output	784x1	784x1	0

Fig. 6. Training curves of the used AE model on the MNIST dataset.

produce the output data which is a copied version of the input. By training the network to perform a copying task, the codes or embeddings are optimized to capture useful properties of the input data. As the result, those embeddings can be used for downstream tasks such as clustering and classification. AE-like structured models have been widely applied to various tasks such as feature extraction [19], image denoising [20], image dehazing [21] [22], and anomaly detection [23].

For the sake of simplicity, in this paper, we adopt a simple AE model with 728-d input/output and 32-d embedding, both the encoder and decoder have 3 hidden layers with ReLU activations. A detailed description of the used AE model is shown in Table I. The AE model is trained to learn the embeddings of MNIST dataset [16]. MNIST dataset contains 60,000 and 10,000 gray-scale images for training and validation, respectively, each image data has a resolution of 28x28 pixels. We flatten all the images to obtain 784-d vectors which are utilized as input of the network.

The processor used in the experiments is Intel(R) Core(TM) i7-4790K CPU @ 4.00GHz. We train the AE model in 100 epochs using the Adam optimizer [24] with a learning rate of 0.001. Several image reconstruction results and the learning curves are shown in Fig. 5 and Fig. 6, respectively. After the model is optimized on the training set, we extract the embeddings from the validation set for the clustering task. As the result, we obtain a set of 10,000 32-d feature embeddings.

IV. Experiments and Results

In order to evaluate the robustness and applicability of the POCS-based clustering algorithm to complex clustering tasks such as feature embedding clustering, we conduct various experiments and analyses to compare its performances in terms of clustering error and execution speed with those of other prevailing clustering methods such as the K-Means and FCM algorithms. In all experiments, each algorithm is executed 20 times, then the mean and standard deviation of clustering error and execution speed are measured and presented. We conduct the comparison experiments under two conditions: *same initial prototypes* and *different initial prototypes*. Specifically, in the first condition, we compare the performance of the POCS-based clustering algorithm with that of the K-Means++ algorithm as these algorithms share a similar prototype initialization procedure. On the other hand, in the second condition, we compare the performances of the POCS-based clustering, K-Means, and FCM algorithms when these algorithms are executed independently.

In addition, classification accuracy is also adopted as an evaluation criterion. Despite the fact that clustering is an unsupervised learning task, the classes can be retrieved when the label information of input data is given. To this end, we determine the class of a cluster as the label that is grouped in that cluster with the highest probability.

A. Condition 1: Same Initial Prototypes

Table II summarizes the performances of the POCS-based and K-Means++ algorithms in terms of clustering error, execution time, and classification accuracy on the Face and MNIST

TABLE II
COMPARISONS OF THE K-MEANS++ AND POCS-BASED CLUSTERING ALGORITHMS IN TERMS OF CLUSTERING ERROR, EXECUTION TIME, AND CLASSIFICATION ACCURACY.

Algorithm	Clustering Error		Execution Time (ms)		Classification Accuracy	
	Face	MNIST	Face	MNIST	Face	MNIST
K-Means++	111.2±0.3	6,804.6±12.8	4.3±0.6	916.9±301.8	99.36±1.7	64.6±2.2
POCS-based	111.3±0.7	6,836.9±27.6	4.1±0.9	771.2±269.6	99.53±1.5	63.9±1.8

TABLE III
COMPARISONS OF THE K-MEANS, FCM, AND POCS-BASED CLUSTERING ALGORITHMS IN TERMS OF CLUSTERING ERROR, EXECUTION TIME, AND CLASSIFICATION ACCURACY.

Algorithm	Clustering Error		Execution Time (ms)		Classification Accuracy	
	Face	MNIST	Face	MNIST	Face	MNIST
K-Means	110.5±0.2	6,692.4±20.1	3.8±2.1	282.1±30.8	91.7±1.4	60.8±0.5
FCM	111.1±0.1	8,323.0±5.2	45.3±6.2	1,484.9±40.2	93.2±1.2	62.7±5.2
POCS-based	111.2±0.2	6,721.5±15.3	4.5±2.3	615.9±243.7	99.2±2.2	64.5±4.2

embedding sets. As summarized in Table II, the POCS-based clustering algorithm can work favorably compared with the K-Means++ algorithm in terms of clustering error even though the K-Means++ algorithm still shows a slightly better result on the MNIST embedding set with a minor gap. On the other hand, in terms of the average convergence speed, the POCS-based clustering algorithm outperforms the K-Means++ algorithm with a minor gap on the Face embedding set (4.1 ms compared with 4.3 ms, respectively) and a significant gap on the MNIST feature embedding set (771.2 ms compared with 916.9 ms, respectively). Additionally, these two clustering algorithms share similar performances in terms of classification accuracy, the POCS-based clustering algorithm slightly outperforms the K-Means++ algorithm on the Face embedding set whereas the K-Means++ algorithm marginally surpasses the POCS-based clustering algorithm on the MNIST feature embedding set.

Based on these comparisons, we empirically conclude that the POCS-based clustering algorithm can perform competitively compared with the K-Means++ algorithm in feature embedding clustering tasks.

B. *Condition 2: Different Initial Prototypes*

Table III summarizes the performances of the K-Means, FCM, and POCS-based clustering algorithms in terms of clustering error, convergence time, and classification accuracy. As can be seen from Table III, the POCS-based clustering algorithm produces a similar performance in terms of clustering error compared to those of the K-Means and FCM algorithms for the Face embedding set, while the FCM algorithm notably shows higher clustering error for the MNIST embedding set. Considering the convergence speed, on the Face embedding set, the POCS-based clustering algorithm obtains the second-best result with 4.5±2.3 ms, which is 10 times faster than the FCM algorithm and slightly slower than the K-Means algorithm. On the embedding set extracted from the MNIST dataset, the POCS-based algorithm also performs with the second-best result and still executes much faster than the FCM algorithm. The main drawback of the POCS-based clustering algorithm here is the instability of the execution time. That is, depending on the initial prototypes, the algorithm may converge extremely fast or slow. Note also that when comparing the K-Means algorithm to the K-Means++ algorithm, the K-Means algorithm can converge much faster on the MNIST feature embedding set due to the difference in prototype initialization procedures. That is, the K-Means algorithm randomly picks the initial prototypes while the K-Means++ algorithm applies a careful seeding method for prototype initialization, and the time consumed for the initializing progress highly depends on the data population. In terms of classification accuracy, the POCS-based clustering algorithm outperforms other methods on both the Face and MNIST embedding sets, while the K-Means and FCM algorithms can produce favorable results and the FCM algorithm achieves a marginally better performance than the K-Means approach.

As a result, the POCS-based clustering algorithm gives a competitive performance when compared with those of the K-Means and FCM algorithms in feature embedding clustering problems. It implies that the POCS-based clustering algorithm has potential in a wide range of clustering tasks.

V. CONCLUSIONS

In this paper, the applicability of the POCS-based clustering algorithm, an effective clustering technique based on the Projection onto Convex Set (POCS) method, to feature embedding clustering problems is examined. The POCS-based clustering algorithm applies the property of the POCS method to clustering problems and has been proven to be able to produce competitive performance compared to other prevailing clustering approaches in terms of clustering error and execution speed. An off-the-shelf FaceNet model and a plain autoencoder network are utilized to synthesize two sets of feature embeddings from the Five Celebrity Faces and MNIST datasets for experiments and analyses. The evaluation results on the synthetic embedding datasets show that the POCS-based clustering algorithm can perform with favorable results and can be considered a promising approach for various data clustering problems.

REFERENCES

[1] J. MacQueen, "Classification and analysis of multivariate observations," in *5th Berkeley Symp. Math. Statist. Probability*. University of California Los Angeles LA USA, 1967, pp. 281–297. 1

[2] L.-A. Tran, N.-P. Le, T.-D. Do, and M.-H. Le, "A vision-based method for autonomous landing on a target with a quadcopter," in *2018 4th International Conference on Green Technology and Sustainable Development (GTSD)*. IEEE, 2018, pp. 601–606. 1

[3] 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

[4] L.-A. Tran, T.-D. Do, D.-C. Park, and M.-H. Le, "Enhancement of robustness in object detection module for advanced driver assistance systems," in *2021 International Conference on System Science and Engineering (ICSSE)*. IEEE, 2021, pp. 158–163. 1

[5] A. Likas, N. Vlassis, and J. J. Verbeek, "The global k-means clustering algorithm," *Pattern recognition*, vol. 36, no. 2, pp. 451–461, 2003. 1

[6] J. M. Pena, J. A. Lozano, and P. Larranaga, "An empirical comparison of four initialization methods for the k-means algorithm," *Pattern recognition letters*, vol. 20, no. 10, pp. 1027–1040, 1999. 1

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

[8] 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

[9] D. C. Park and I. Dagher, "Gradient based fuzzy c-means (gbfcm) algorithm," in *Proceedings of 1994 IEEE International Conference on Neural Networks (ICNN'94)*, vol. 3. IEEE, 1994, pp. 1626–1631. 1

[10] L. M. Bregman, "The relaxation method of finding the common point of convex sets and its application to the solution of problems in convex programming," *USSR computational mathematics and mathematical physics*, vol. 7, no. 3, pp. 200–217, 1967. 1, 2

[11] R. Y. Albert, R. J. Marks II, K. E. Schubert, C. Baylis, A. Egbert, A. Goad, and S. Haug, "Dilated pocs: Minimax convex optimization," *IEEE Access*, 2023. 2, 3

[12] 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. 2, 3

[13] L. Yang, W. Fan, and N. Bouguila, "Clustering analysis via deep generative models with mixture models," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 33, no. 1, pp. 340–350, 2020. 2

[14] 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. 2, 3

[15] "Five celebrity faces dataset," https://www.kaggle.com/datasets/dansbecker/5-celebrity-faces-dataset. 2, 3

[16] 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. 2, 3, 4

[17] 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. 3

[18] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning internal representations by error propagation," California Univ San Diego La Jolla Inst for Cognitive Science, Tech. Rep., 1985. 3

[19] N. Lv, C. Chen, T. Qiu, and A. K. Sangaiah, "Deep learning and superpixel feature extraction based on contractive autoencoder for change detection in sar images," *IEEE transactions on industrial informatics*, vol. 14, no. 12, pp. 5530–5538, 2018. 4

[20] A. Majumdar, "Blind denoising autoencoder," *IEEE transactions on neural networks and learning systems*, vol. 30, no. 1, pp. 312–317, 2018. 4

[21] L.-A. Tran, S. Moon, and D.-C. Park, "A novel encoder-decoder network with guided transmission map for single image dehazing," *Procedia Computer Science*, vol. 204, pp. 682–689, 2022. 4

[22] L.-A. Tran and D.-C. Park, "Encoder–decoder network with guided transmission map: Robustness and applicability," in *International Symposium on Intelligent Informatics*. Springer, 2022, pp. 41–54. 4

[23] D. L. Aguilar, M. A. Medina-Pérez, O. Loyola-Gonzalez, K.-K. R. Choo, and E. Bucheli-Susarrey, "Towards an interpretable autoencoder: A decision-tree-based autoencoder and its application in anomaly detection," *IEEE transactions on dependable and secure computing*, vol. 20, no. 2, pp. 1048–1059, 2022. 4

[24] D. P. Kingma and J. Ba, "Adam: A method for optimization," *arXiv preprint arXiv:1412.6980*, 2014. 4