applied sciences

MDPI

Article

Fuzzy Rough C-Mean Based Unsupervised CNN Clustering for Large-Scale Image Data

Saman Riaz [1,*], Ali Arshad [1] and Licheng Jiao [2]

[1] School of Computer Science and Technology and School of International Education, Xidian University, Xi'an 710071, China; alli.arshad@gmail.com

[2] School of Artificial Intelligence, Key Laboratory of Intelligent Perception and Image Understanding of Ministry of Education of China, International Research Center of Intelligent Perception and Computation and International Joint Collaboration Laboratory of Intelligent Perception and Computation, Xi'an 710071, China; lchjiao@mail.xidian.edu.cn

* Correspondence: samanriaz@hotmail.com

Received: 14 August 2018; Accepted: 5 October 2018; Published: 10 October 2018

Abstract: Deep learning has been well-known for a couple of years, and it indicates incredible possibilities for unsupervised learning of representations with the clustering algorithm. The forms of Convolution Neural Networks (CNN) are now state-of-the-art for many recognition and clustering tasks. However, with the perpetual incrementation of digital images, there exist more and more redundant, irrelevant, and noisy samples which cause CNN running to gradually decrease, and its clustering accuracy decreases concurrently. To conquer these issues, we proposed an effective clustering method for a large-scale image dataset which combines CNN and a Fuzzy-Rough C-Mean (FRCM) clustering algorithm. The main idea is that first a high-level representation, learned by multi-layers of CNN with one clustering layer, produce the initial cluster center, then during training image clusters, and representations, are updating jointly. FRCM is utilized to update the cluster centers in the forward pass, while the parameters of proposed CNN are updated by the backward pass based on Stochastic Gradient Descent (SGD). The concept of the rough set of lower and boundary approximations deal with uncertainty, vagueness, and incompleteness in cluster definition, and fuzzy sets enable efficient handling of overlapping partitions in the noisy environment. The experiment results show that the proposed FRCM based unsupervised CNN clustering method is better than the standard K-Mean, Fuzzy C-Mean, FRCM and also other deep-learning-based clustering algorithms on large-scale image data.

Keywords: unsupervised clustering; convolution neural network; fuzzy learning; rough sets

1. Introduction

Image clustering [1–13] is a consequential research field in image processing and computer vision applications. Nowadays, it has come into an age of big data (with many novel and portable digital devices). Every day millions of data are produced and worldwide a large number of digital images are uploaded to the cloud for storage or sharing. With more and more data being generated, efficiently consolidating the large-scale image data is becoming a demanding problem. However, many researchers focus on decreasing data from a large-scale dataset [14] and feature encoding [15] for large-scale image clustering. The reason for this is the existence of many redundant and noise samples among the entire large data [16].

Unsupervised clustering is an essential machine-learning technique that is utilized to discover the common natural cluster structure of an unlabeled dataset [17]. Therefore, a great clustering

algorithm ought to depend as little as possible on earlier learning, which is typically not available before cluster analysis.

Distinctive clustering techniques such as hierarchical clustering [18–20], centroid-based clustering [1–5,21,22], graph theoretical approaches [23,24], and the density-based approach [25] have been broadly applied for image clustering. However, with the data quality increasing, another basic issue of data uncertainty unavoidably occurs: the big data themselves may contain a high measure of uncertainty, vagueness, and overlapping among clusters that is therefore a huge challenge to the clustering task.

To overcome the above issues, two main mathematical tools of granular computing [26], rough set theory and fuzzy set theory, have been introduced into the clustering method to expose uncertainties among the raw data [27–31], which expand the clustering from hard-partition to soft-partition and in this manner shape an essential branch of the data clustering [32–34]. For example, Fuzzy C-Mean (FCM) clustering [35,36], Rough C-Mean (RCM) clustering [37,38], Hybrid Rough-Fuzzy clustering [27], and Fuzzy-Rough C-Mean (FRCM) clustering [39]. FCM is one prominent clustering algorithm generally utilized in the applied field by appointing a fuzzy membership to each sample and presenting a fuzziness weighting exponent. RCM describes a cluster by a center and pair of lower and upper approximations, which are weighed against diverse parameters in computing the new center in the algorithm. FRCM is the incorporation of rough set theory and fuzzy set theory in which each cluster is represented by a center, a crisp lower approximation and a fuzzy boundary in this algorithm, and the new center is a weighting average of the boundary and lower approximation.

In our paper, we introduce FRCM-based unsupervised convolution neural network (CNN) clustering. A key component of our approach is the reliable noise-free samples that are selected by FRCM algorithm, during updating of cluster centers. The concept of fuzzy sets enables efficient handling of overlapping partitions and rough sets deal with the uncertainty of robust data.

Deep learning (DL) has become a hot topic in the last couple of years. Deep learning is a developing and intense model that permits large-scale task-driven features to learn from big data. These improvements were obtained with supervised learning, whereas our goal is unsupervised image data clustering.

A lot of remarkable deep-learning-based architectures have been proposed recently, such as the Deep Belief Network, Deep Boltzmann Machine, Deep Autoencoder, and Convolutional Neural Network (CNNs). However, the most well-known network architectures for image/video applications is CNN, including CNN based AlexNet [40], ResNet [41], VGG (Visual Geometry Group) [42], and FCN (Fully Convolutional Networks for semantic segmentation) [43]. Since they were introduced into deep learning, the CNNs have demonstrated a state-of-the-art accuracy in large-scale image classification. CNN based architectures play an important role in the processing of image data due to their unique structure in the phase of feature representation, given a sufficiently large labeled training set.

The performance of clustering greatly depends on (1) noise reduction and (2) feature representational power. There have been few attempts to increase the performance of image clustering by deep-networks-based feature representation learning. Nevertheless, deep-networks-based feature representational learning does depend on much labeled training data, which is not accessible in unsupervised clustering. To overcome this problem, the model can be pre-trained in view of existing large-scale training image sets; the pre-trained model cannot fit the normal input data divider.

Various works have investigated consolidating image clustering with representation learning [44–47]. Hsu [44] proposed CNN-based joint clustering in which Mini-batch K-Mean is then executed to assign cluster labels. However, it is a completely deterministic model that sheds no light on data uncertainty reduction. Xie [45] proposed Auto-Encoder based deep-learning, figuring out how to learn visual representations followed by conventional K-Mean to get the final cluster. However, for high-dimensional data, the Auto-Encoder generally cannot learn representative features appropriately. Dundar (CNN-CM) [46] and Yang (CNN-RC) [47,48] proposed CNN-based clustering with connection Matrix and re-running clustering respectively. However, when the size of the image

data is large then it devours high computation and memory complexity. Instead, an unnecessary number of input data would increase the parameter fine-tuning frequency of CNNs; this could cause the risk of uncertainties and over-fitting among the raw data [49,50]. Therefore, it is normal to reason that there must be considerable noise and redundant data in large-scale data sets.

To overcome the above issues, the contribution of our work is that we can decrease the training time and furthermore maintain and even improve the test accuracy by selecting noise-free data for updating clusters by using the Fuzzy-Rough C-Mean (FRCM) algorithm. In this paper, we satisfy the model with unsupervised CNN clustering and FRCM algorithm. Our proposed clustering algorithm benefits from both CNNs and FRCM by merging one with the other; this is shown in Figure 1. Unsupervised CNN clustering (UCNN) architecture is proposed, which can extract salient features to produce the initial cluster centers. During the learning stage the cluster and representation are updating simultaneously: cluster centers are updating based on robust free samples by FRCM algorithm in forward-pass and the parameters of representation are updating in backward-pass on stochastic gradient descent (SGD). The main idea behind our method is that good representations are beneficial to image clustering and better clustering results beneficial for feature representations.

The main contributions of the proposed algorithm are as follows:

1. We present an FRCM-based unsupervised CNN clustering, which is robust to the uncertainties in the training data and could achieve reliable performances even with noisy samples.
2. We propose a joint learning framework to simultaneously update the parameter of unsupervised CNN and the cluster centroid iteratively.
3. We introduce FRCM with CNN to reduce the time complexity and increase cluster performance by updating the cluster centers based on a reliable sample selection, which is a key component of our method to ensure its success.
4. Extensive experiments on large-scale image datasets indicate that our strategy to enhance the clustering accuracy, when compared with other non-fuzzy deep neural networks, show that fuzzy learning is without a doubt a conceivable method to further improve the performance of deep-learning-based clustering algorithms.

This paper is organized as follows: Section 2 deals with the implementation strategy of the algorithm, Section 3 describes the experiment and results, Section 4 provides the threats to validity and Section 5 provides the conclusion.

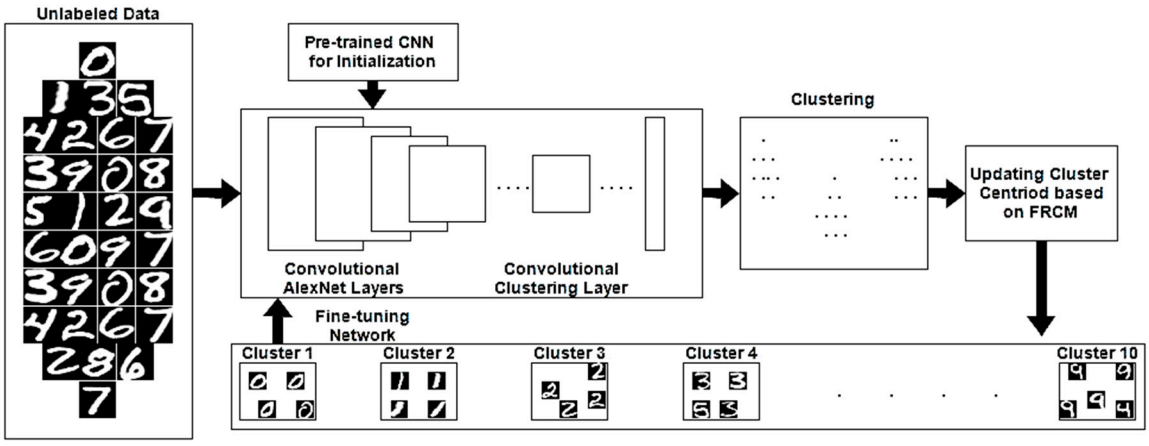


Figure 1. Flowchart of the proposed clustering method for joint representation learning and image clustering.

2. Fuzzy Rough C-Mean Based Unsupervised CNN Clustering

2.1. The Problem of Deep-Learning-Based Clustering

In recent years, the clustering method with the deep neural network has gained additional interest due to the success of supervised learning [46,48,51–68]. However, in most cases, clustering is handled in the unsupervised form, making its application with deep learning more significant and requiring more modeling effort and theoretical analysis.

Previously deep learning-based clustering algorithms have not focused on data uncertainty reductions to increase the accuracy and decrease the time complexity.

Xie et al. [45] proposed the first deep-learning based clustering. Deep Embedded Clustering (DEC) is based on auto-encoders as network architecture followed by conventional K-Means for final clustering. The network model is fine-tuned using the cluster assignment hardening less and the cluster centers are updated.

Yang et al. [64] also proposed the auto-encoder-based method followed by K-Mean clustering. However, the network is jointly trained using a combination of the representation learning and image clustering.

Lie et al. [58] proposed an idea almost identical to DEC except for using a conventional auto-encoder. However, for high-dimensional data, the auto-encoder usually cannot learn representation features well compared to CNN-based architectures.

Yang et al. [47] proposed a convolution neural network with re-running clustering (CNN-RC) method. For clustering, a hierarchical clustering approach is utilized. Concerning the training part, the network is jointly trained, and the cluster is updated in the forward pass, while representation learning is in the backward pass. However, compared the centroid-based clustering, hierarchical clustering devours high computation and memory complexity due to the size of image data which becomes large.

Dundar [46] proposed a CNN with the connection matrix (CNN-CM) method; for clustering K-Mean is utilized. A connection matrix is proposed that enables encouraging in additional side data to help to learn the representation for clustering. In the view of learned features, a full-set K-Mean is then performed to gather all images into their relating clusters. However, when the size of data becomes large, the complexity of full-set K-Mean will increase.

Hu & Lin [44] proposed clustering CNN to achieve joint clustering and representation learning with feature Drift Compensation for large-scale image data. They extracted the silent features from one of the internal layers of CCNN. At first, initial cluster centroids are assumed from extracted features of randomly picked k samples, and K-Mean is performed on the features extracted from the input dataset to get corresponding cluster labels. Based on the assigned labels and labels predicted by the Softmax layer, the network parameters can be updated. Further, the corresponding cluster centroids are updated by extracted features of the mini-batch. However, the fuzzy modeling achieves many advantages over the non-fuzzy method, such as robustness against uncertainties, vagueness, and overlapping dataset.

2.2. Background of Fuzzy Rough C-Mean (FRCM)

Hu et al. [69] proposed a FRCM clustering algorithm, which is the development and combination of FCM [70] and RCM [37,38]. As we know, FCM maps a membership over the range 0–1; each object belongs to some or all of the clusters to some fuzzy degree. RCM classifies the object space into three parts: lower approximation, boundary, and negative region. All the objects with RCM in the lower approximation take the same weight and all the objects in the boundary take another weighting index uniformly. In fact, the objects in the lower approximation definitely belong to a cluster, but the objects in the boundary regions belong to a cluster to some extent and have different influences on the centers and clusters, so different weighting values should be imposed on the boundary objects in computing the new centers. Inspired by the above idea, FRCM [69] integrates the advantage of fuzzy

set theory and rough set theory and incorporates fuzzy membership values of each sample to the lower approximation and boundary area of a cluster. Let a set of image data $I = \{I_1, I_2, \ldots, I_{n_x}\} \in \mathrm{R}^d$, where d is the dimension of the data points. Each cluster C_j $(j = 1, 2, \ldots, k)$ is regarded as a rough set. It is categorized by the lower approximations $\underline{C_j}$, the upper approximations $\overline{C_j}$ and the boundary area $C_j^{\mathrm{B}} = \overline{C_j} - \underline{C_j}$, respectively. Let $c = \{c_1, c_2, \ldots, c_k\}$ be a vector composed of k centers of clusters, where $c_{\mathrm{j}} \in \mathrm{R}^d$. The objects in lower approximation belong to a cluster categorically, however the objects in the boundary regions belong to a cluster to some extent and have diverse effect on the centers and clusters, so different weighting values ought to be imposed on the boundary objects in computing the new centers.

Let $u = \{u_i(j)\}_{n_x \ x \ k}$ be a membership matrix, we can define that membership function as follows:

$$u_{ij} = \begin{cases} 1, \\ \dfrac{1}{\sum_{s=1}^{k} (d_{ij}/d_{sj})^{2/(m-1)}} \end{cases}, \quad i = 1, 2, \ldots, n_x, \quad I_i \in \underline{C_j} \quad I_i \in C_j^{\mathrm{B}}, \quad j = 1, 2, \ldots, k \tag{1}$$

The exponent $m > 1$ is utilized to change the weighting impact of membership values.
The new cluster center is computing as follows

$$c_j^{(l+1)} = \frac{\sum_{i=1}^{n_x} \left(u_{ij}^{(l)}\right)^m \mathrm{I}_i}{\sum_{i=1}^{n_x} \left(u_{ij}^{(l)}\right)^{\mathrm{m}}}, \ j = 1, \ 2, \ \ldots, \ \mathrm{k} \tag{2}$$

The objective function of FRCM is

$$\mathrm{J}_m^{(l)}(u, c) = \sum_{i=1}^{n_x} \sum_{j=1}^{\mathrm{k}} (u_{ij})^m \| I_i - c_j \|^2 \tag{3}$$

The FRCM can be formulated as follows.

Input: Unlabeled data I, number of cluster k, threshold parameter T, exponent index m, stop criterion ε.
Output: membership matrix u, k cluster centers.

Step 0: Let $l = 0$, initialization the centers $\mathrm{c}_j^{(l)}$, $j = 1, \ 2, \ \ldots, \ k$ using random sampling.

Step 1: Assign the data objects to the approximations

i. For a given data object I_i calculate its closest center $\mathrm{c}_h^{(l)}$ and $\mathrm{A}^{(l)}$ as follows

$$d^{(l)} = dist\left(I_i, \ \mathrm{c}_h^{(l)}\right) = \min_{1 \le j \le \mathrm{k}} dist\left(I_i, \ \mathrm{c}_j^{(l)}\right) \tag{4}$$

$$\mathrm{A} = \left\{ \forall \, s, \ s = 1, 2, \ldots, k, \ s \neq \mathrm{h} : \ \mathrm{d}_{is}^{(l)} - \mathrm{d}_{ih}^{(l)} \le \mathrm{T} \right\}$$

ii. If $\mathrm{A} \neq \varnothing$, then $I_i \in \overline{\mathrm{C}_h}$, $I_i \in \underline{\mathrm{C}_s}$, and $\mathrm{I}_i \notin \underline{\mathrm{C}_l}$, l = 1, 2, . . . , k.
If $\mathrm{A} \neq \varnothing$, then $I_i \in \underline{\mathrm{C}_h}$, $I_i \in \overline{\overline{\mathrm{C}_h}}$.

Step 2: Compute membership values using Equation (1).
Step 3: Compute new cluster center by using Equation (2).

Step 4: Check convergence of the algorithm. If the algorithm has converged, stop, else $l = l + 1$ go to Step 1.

2.3. FRUCNN Clustering Architecture

To enhance the performance of the clustering algorithm, we integrate unsupervised CNN (UCNN) [71] with Fuzzy Rough C-Mean clustering to the proposed new clustering algorithm, which is shown in Figure 1. It is generally divided into two parts such as the pre-clustering part and further joint clustering and representation learning. During the learning stage, the clusters are updated by the FRCM algorithm. The pre-clustering part, Figure 2 show the network architecture of our proposed unsupervised convolutional neural network which contains multi-convolution with one clustering layer. During the pre-clustering part, the size of multi-convolution layers depends on the size of the dataset. For big data, it needs large-scale networks. For example, for image net [72] the unsupervised CNN clusters consist of five convolutional layers received from the initial five convolutional layers (Conv 1–Conv 5) of AlexNet [40], followed by three adjustment layers (Conv 6, Conv 7, and CConv) with channel number 6144, 2048 and k, respectively, that supplant the fully connected (FC) layer in AlexNet. The adjustment layers (Conv 6, Conv 7, and CConv) involve two convolutional layers (Conv 6–Conv 7) with one clustering convolutional layer (CConv) with k clusters, all with 3×3 kernels followed by global max-pooling. The maximum value for each channel of the clustering convolutional layer (CConv) is the output of the max-pooling so that the size is $1 \times k$. Finally, we join with a fully connected layer (FC) and Softmax layer to extract the image features.

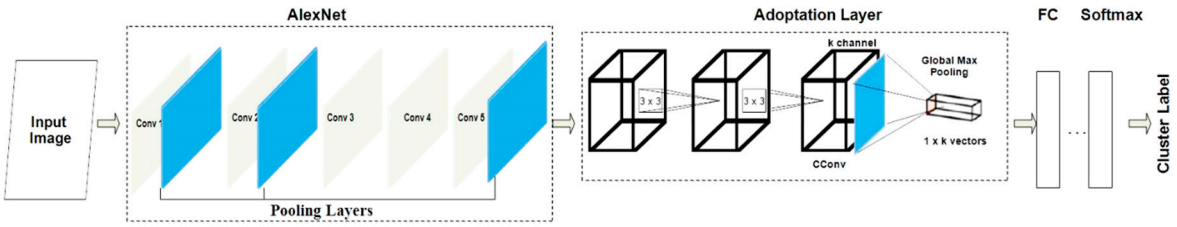


Figure 2. A proposed unsupervised Convolution Neural Network (CNN) based network architecture, that consists of few convolutional layers, followed by some adjustment to convolution layers with one clustering convolution layer (CCnov), a fully connected layer, and Softmax layer.

2.4. Joint Clustering and Representation Learning

Suppose the unlabeled dataset contains n_x images $I = \{I_1, I_2, \ldots, I_{n_x}\}$. The main objective is to group n_x images into k clusters $C = \{C_1, C_2, \ldots, C_k\}$. Let $H = \{h_1, h_2, \ldots, h_{n_x}\}$ be the set of extracted features from the FC layer of UCNN using filters $h_i = f(W_{FC}/I_i)$, where W_{FC} represents the set of parameters (weights) of FC layer. We use FRCM to update the clusters by using features extracted from the FC layer as initial cluster centers.

In our proposed method, given an input image set, we first randomly pick k samples and extract their features as an initial cluster centroid using the proposed UCNN with an initial pre-trained image from the ImageNet datasets. FRCM is then performed to assign cluster labels to individual images randomly sampled from the input set until all images are processed. Subsequently, the proposed UCNN simultaneously updates the parameters of proposed UCNN and the centroids of image clusters iteratively based on stochastic gradient descent.

In the learning part, the weight W_{FC} and cluster centroid c will be updated simultaneously using Algorithm 2, the cluster centroid is updated by using FRCM using Algorithm 1, and updating the representation parameters by stochastic gradient descent (SGD).

2.4.1. Pre-Processing Data for UCNN

Data argumentation is utilized to increase sample variety during the initial pre-clustering process. After the initialization, we used ILSVRC12 training set of ImageNet [73] to pre-train the parameters of Conv 1–Conv 5 in the AlexNet [40].

2.4.2. Cluster Centroid Updating

Let $I = \{I_1, I_2, \ldots, I_{n_x}\}$ be the set of n_x images. Initially, k random images are selected from input image set I and extract their features H_j using the pre-trained UCNN $H_j^{(t)} = \{h_1^t, h_2^t, \ldots, h_k^t\} \in H_{\mathrm{FC}}$ from FC layer as initial cluster centroid c in the initial iteration (i.e., t = 0) by UCNN network. The Fuzzy-Rough C-Mean (FRCM) algorithm [39] is performed to update the cluster centroid by objective function of FRCM, which is

$$J_m(u,h) = \sum_{i=1}^{n_x} \sum_{j=1}^{k} (u_{ij})^m \|I_i - h_j\|^2 \tag{5}$$

The exponent $m > 1$ is utilized to change the weighting impact of membership values.
Updated cluster centroid by FRCM is

$$h_j^{t+1} = c_j^{t+1} = \frac{\sum_{i=1}^{n_x} (u_{ij}^t)^m I_i}{\sum_{i=1}^{n_x} (u_{ij}^t)^m},\ j = 1, 2, \ldots, k \tag{6}$$

In iteration t, the jth-centroid c_j^t that is assigned as a new sample $h_j^{(t)}$, where $h_j^{(t)}$ represent the extracted features for FC layer.

Algorithm 1 The Cluster Centroid Updating

Input:
Unlabeled I_{n_x} image dataset,
Number of cluster k,
Randomly select k image I_{nk} from I_{nx},
Extract image feature from I_{nk} images as the initial centroid.
Initial cluster center c, at t = 0,
Exponent index m,
Threshold parameter T,
Stop criterion ε.
Output:
Updated cluster centroid c,
Updated extracted features for FC layer.

1. Let t = 0, initialization the cluster center
   $c_j^{(o)} = h_j^{(o)} = \{h_1^o, h_2^o, \ldots, h_k^o\} \in H_{\mathrm{FC}}^o,\ j = \{1, 2, \ldots, k\}$.
2. Assign the data object to the approximation by Equation (4).
3. Compute $u_{ij}^{(t)}$ according to Equation (1).
4. Compute new cluster center c_j^{t+1} according to Equation (6).
5. Check convergence of the algorithm. If the algorithm converged, stop, else $t = t + 1$ go to step 2.
6. Assigned features for FC layer as
   $h_j^{(t+1)} = c_j^{(t+1)}$.
7. End

2.4.3. Representation Learning

By using unsupervised CNN, the features are extracted for FC layers to generate the features for clustering by the output of the extracted local salient features from CConv layer [74]. To learn the parameters $\theta(w_{ri},\ w_{ij})$ of FC and softmax layers of UCNN, we utilize SGD process [75] as in Figure 3, where w_{ri} is the set of weights of layer FC and w_{ij} is the set of weights of softmax. In order to learn the parameters of layers FC and softmax, we first define the objective function.

$$L = \frac{1}{2}\sum_{j=1}^{k}(\hat{y}_j - y_j)^2 \tag{7}$$

where k is denoted as the number of the cluster, $\hat{y}_j$ is the predicted jth cluster label utilizing UCNN and y_j is the predicted jth cluster label by using FRCM that is used as a pseudo ground-truth to assist the update of the UCNN clustering model. Then we compute the gradient of the objective function w.r.t w_{ri} for updating the weights of FC. For this, first use the chain rule to calculate gradient w.r.t w_{ij} as follows.

$$\frac{\partial L}{\partial w_{ij}} = \frac{\partial L}{\partial \hat{y}_j}\cdot\frac{\partial \hat{y}_j}{\partial u_j}\cdot\frac{\partial u_j}{\partial w_{ij}} \tag{8}$$

where u_j is the activation function of the jth ReLU [76]. The partial derivative of L w.r.t $\hat{y}_j$ is

$$\frac{\partial L}{\partial \hat{y}_j} = \hat{y}_j - y_j \tag{9}$$

and the partial derivative of ReLU w.r.t its u is

$$\frac{\partial y_j}{\partial u_j} = \max(\hat{y}_j,\ 0) \tag{10}$$

The partial derivative of $u_j = \sum_{i=1}^{k} w_{ij}h_i$ w.r.t w_{ij} is

$$\frac{\partial u_j}{\partial w_{ij}} = h_i \tag{11}$$

Consequently, w_{ij} can be updated in the tth iteration by

$$w_{ij}^{(t+1)} = w_{ij}^{(t)} - \eta\left(\hat{y}_j - y_j\right)\cdot\max\left(\hat{y}_j,\ 0\right)\cdot h_i^{(t)} \tag{12}$$

where η is the learning rate. Now by using chain rule to calculate gradient w.r.t w_{ri} as follows.

$$\frac{\partial L}{\partial w_{ri}} = \sum_{j=1}^{k}\left(\frac{\partial L}{\partial \hat{y}_j}\cdot\frac{\partial \hat{y}_j}{\partial u_j}\cdot\frac{\partial u_j}{\partial h_{i\ new}}\right)\cdot\frac{\partial h_i}{\partial u_i}\cdot\frac{\partial u_i}{\partial w_{m\ i}} \tag{13}$$

where $\frac{\partial u_j}{\partial h_i} = w_{ij}$, the derivative of ReLU h_i w.r.t u_i is

$$\frac{\partial h_i}{\partial u_i} = \max(h_i\ , 0) \tag{14}$$

and $\frac{\partial u_i}{\partial w_{r\ i}}$ is

$$\frac{\partial u_i}{\partial w_{r\ i}} = \frac{\partial \sum_{r=1}^{k} w_{r\ i}\ x_r}{\partial w_{r\ i}} = x_r \tag{15}$$

Consequently $w_{r\,i}$ can be updated in the tth iteration by

$$\begin{aligned} w_{r\,i}^{(t+1)} &= w_{r\,i}^{(t)} - \eta \sum_{j=1}^{k} \left[\left(\hat{y}_j - y_j\right) \cdot \max\left(\hat{y}_j,\, 0\right) \cdot w_{ij} \right] \cdot \max\left(h_i^t, 0\right) \cdot x_r \\ &= w_{r\,i}^{(t)} - \eta\, \Delta\, w_{r\,i}^{(t)} \end{aligned} \tag{16}$$

Equation (16) is the full gradient for updating the weights of FC layer.

Algorithm 2 Joint Cluster Centroid Updating and Representation Learning

Input:
Input image dataset I,
Randomly select k images dataset I_{nk},
Number of cluster k,
Learning rate η,
Max iteration τ,
Randomly pick k images from I and extract image features h_j.
Then initial cluster centers, $c_j^{(o)} = h_j^{(o)} \in H_{FC}^o$
Output:
Final cluster centroid,
Final weight (w_{ri}, w_{ij})

1. For t = 1 to τ to do.
2. Calculate y_j cluster label using FRCM [39] as a ground-truth.
3. Update cluster centroid by using Algorithm 1.
4. Find predicted cluster label $\hat{y}_j$ using updated cluster centroid in FC layer of UCNN network.
5. Update weight (w_{ri} and w_{ij}) by using Equations (12) and (16).
6. Fine tune (UCNN, k, $\hat{y}$) by using Equation (7).
7. End.

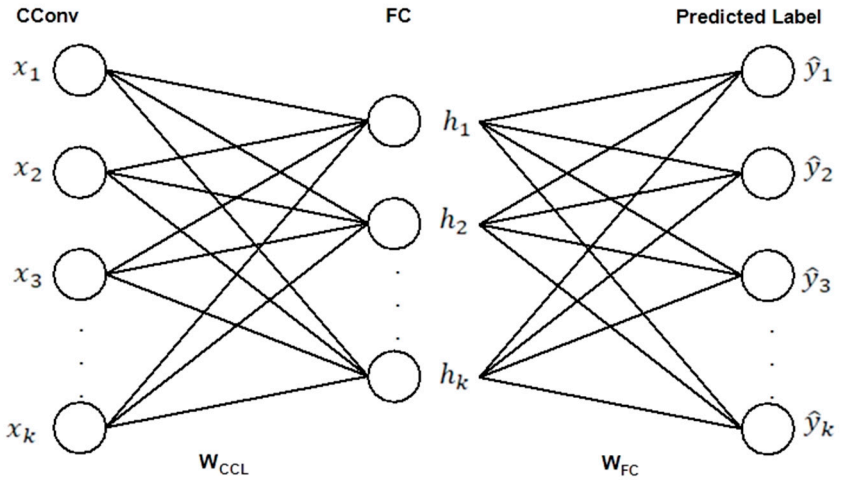


Figure 3. Flowchart of updating the parameters of fully connected (FC Layer) and Softmax layers.

3. Experiments

3.1. Data Preparation

In this paper, MATLAB 2018a [77] was utilized as the programming tool. Table 1 is the description of three publically available datasets on which the experiments were performed. We selected one large-scale image dataset, ILSVRC12 in ImageNet [73], which consists of 1.2 million training images and

50 thousand validation images of 256 × 256-pixel size collected from one thousand object categories. Other than the large-scale image dataset, we additionally evaluated the performance of our proposed approach on two smaller scale dataset; MNIST [78] and Youtube–Face (YTF) [79]. MNIST contains 60 thousand training images and 10 thousand testing images of hand-written digits of 28 × 28-pixel size; the digits are centered and the size is normalized. YTF consists of 10 thousand images 55 × 55-pixel size. The images are cropped faces and then resized into a constant. The personal computer is equipped and is used for experiments on all dataset with a commercial GPU card.

Table 1. Description of Datasets.

Dataset	No of Sample	No. of Classes	Image Size
ILSVRC12	1,250,000	1000	256 × 256
MNIST	70,000	10	28 × 28
YTF	10,000	41	55 × 55

3.2. Performance Measure

We adopted four widely utilized clustering performance measures to evaluate the performance of the proposed method, Normalized Mutual Information (NMI) [80], Clustering Accuracy (ACC) [81], Mean of F-Measure (MFM), and Mean of Area Under the Curve (MAUC) [82].

NMI is defined as

$$\mathrm{NMI}(\mathrm{C},\mathrm{Y}) = \frac{\mathrm{I}(\mathrm{C},\mathrm{Y})}{\sqrt{\mathrm{H}(\mathrm{C})\mathrm{H}(\mathrm{Y})}} \tag{17}$$

where C is the class label, Y is the cluster label, H(.) stands for the Entropy and $\mathrm{I}(\mathrm{CY}) = \mathrm{H}(\mathrm{C}) - \mathrm{H}(\mathrm{C}/\mathrm{Y})$ denotes the mutual information between C and Y. A higher NMI is more consistent for clustering results.

The Acc gives the same weight for each class. The final result is obtained by the average value of the accuracy rate of each class independently. Acc is defined as.

$$\mathrm{Acc} = \frac{\sum_{i=1}^{n} \mathrm{Acc}_i}{m} \tag{18}$$

where m is the number of classes and Acc_i stands for the accuracy rate for the ith class.

Binary class problems are shown in Table 2 and donate categorized results of True and False. Here, the minority class is considered positive and the majority class is considered negative. Several measures can be deduced from the confusion matrix for the binary class problem.

$$\mathrm{Recall} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}} \tag{19}$$

$$\mathrm{Precision} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}} \tag{20}$$

$$\mathrm{F-Measure} = \frac{2 \times \mathrm{recall} \times \mathrm{Precision}}{\mathrm{recall} + \mathrm{Precision}} \tag{21}$$

The mean F-Measure can be defined for multi-class problems as follows:

$$\mathrm{MFM} = \frac{\sum_{i=1}^{m} (\mathrm{FM})_i}{m} \tag{22}$$

where m is the total number of classes and i is the index for positive class.

Mean AUC is the average of the pairwise AUC values of all pairs of classes which is defined as

$$\begin{aligned} \text{MAUC} &= \frac{2}{m(m-1)} \sum_{i<j} (\text{AUC}\ (C_i, C_j)) \\ &= \frac{2}{m(m-1)} \sum_{i<j} [A(C_i, C_j) + A(C_j, C_i)] \end{aligned} \tag{23}$$

For two classes C_i & C_j, the value of AUC (C_i, C_j) represents the probability of being assigned to the ith class by the classifier. A randomly selected sample from the first class (ith class) has a higher probability to assign compared to a randomly selected sample from the second class (jth class) and vice versa.

Table 2. Confusion Matrix.

Dataset	Predicted Positive Class	Predicted Negative Class
Actual Positive Class	True Positive (TP)	False Negative (FN)
Actual Negative Class	False Positive (FP)	True Negative (TN)

3.3. Comparison Schemes

To analyze the performance of our approach on large-scale image dataset, we compared our model into two parts. In the first part, we tested our method with seven clustering models with two state-of-the-art clustering models including K-Mean [83] and Fuzzy C-Mean [84]; also five deep learning based clustering models including Deep Embedded Clustering (DEC) [45], Deep Embedded Regularized Clustering (DEPICT) [53], Convolution Clustering for Unsupervised Learning (CNN-CM) [46], Joint Unsupervised Learning of Deep Representations and image cluster (JULE) [47] and CNN-Based Joint Clustering and Representation learning (CCNN) [44]. Besides the seven cluster models, in the second part we applied three baseline schemes for performance evaluation. Scheme 1: the proposed method without update cluster centroid by FRCM. Scheme 2: the proposed method without iterative representation learning using updated cluster centroid. Scheme 3: the proposed method with update cluster centroid and iterative representation learning simultaneously.

3.4. Implementation Details

We utilized AlexNet [40] pre-trained on the ILSVRC12 training set on ImageNet as our basic CNN models to avoid turning any hyper-parameters utilizing the labeled data and also to accelerate the convergence. Data augmentation was utilized to increase the sample variety during the pre-training process. We considered that the number of convolutional layers depends on the size of the image in the dataset. We selected ILSVRC12 in ImageNet [73] as large-scale image dataset. We demonstrated the performance of our clustering method on ILSVRC12 validation set denoted as "ILSVRC12-Val" and did not evaluate theILSVRC12 training set for fairness. The proposed unsupervised CNN clustering for ImageNet is composed of five convolutional layers assumed from AlexNet [40], followed by two adjustment layers with channel size 6144 & 2048 with 3×3 kernel size and one clustering convolutional layer with k channel size with 3×3 kernels followed by a global max-pooling with $(1 \times k)$ output, where k is denoted as the number of clusters that replace the one fully connected layer and softmax layer with k channel size.

We also evaluated the performance of our approach on two other image dataset: MNIST and YTF. The image size of these datasets is substantially smaller than the ImageNet. So, we composed two convolution layers adopted form AlexNet followed by one clustering convolution layer with one fully connected (FC) layer and the other Softmax layer. The output of FC layer is considered the initial centroid of clusters which is updated by utilizing the Fuzzy Rough C-Mean (FRCM) algorithm in the forward pass and representation learning of cluster convolution, FC and softmax layers by backward pass using SGD. The personal computer is utilized to compute all results with a commercial GPU card.

3.5. Experimental Design

The experimental results analysis is mainly split into two parts. In the first part, FRUCNN is tested on three benchmark large-scale image datasets with comparison to the other state-of-the-art clustering methods and deep-learning-based clustering methods, and in the second part, we will discuss three different schemes of FRUCNN and experimentally compare their performances in the consequent test.

Table 3 shows the results of the performance measure (NMI, Acc, MFM, and MAUC) on four datasets to demonstrate the effectiveness of our proposed approach. From the analysis of the results of all performance measures, MNIST-Full and MNIST-Test show better results compared to ILSVRC12-Val, and YTF. For further analysis of the results of large-scale dataset, we compared our proposed approach to other state-of-the-art methods.

Table 3. Results of Performance Measures (NMI, Acc, MFM & AUC) of Proposed work (FRUCNN).

Dataset	NMI (Normalized Mutual Information)	ACC (Clustering Accuracy)	MFM (Mean of F-Measure)	MAUC (Mean of Area Under the Curve)
ILSVRC12	0.401	0.456	0.400	0.412
MNIST-Full	0.919	0.971	0.870	0.924
MNIST-Test	0.920	0.974	0.910	0.945
YTF	0.891	0.783	0.650	0.736

FRUCNN, Fuzzy Rough C-Mean Based Unsupervised CNN Clustering.

In Table 4, the reported results are borrowed from the original paper; for those that did not show the results, the corresponding results are found by re-running the code released by original papers. We put dash marks (-) for the results that are not applied to obtain.

Table 4. Comparison between the Different Clustering Algorithms Based on Clustering Performance (NMI & ACC).

Methods	ILSVRC12-Val		MNIST-Full		MNIST-Test		YTF	
	NMI	ACC	NMI	ACC	NMI	ACC	NMI	ACC
K-Mean	-	-	0.500	0.534	0.501	0.547	0.776	0.601
FCM	-	-	0.613	0.657	0.615	0.660	0.779	0.632
DEC	0.155	-	0.816	0.844	0.827	0.859	0.446	0.371
DEPICT	0.371	-	0.917	0.965	0.915	0.963	0.802	0.621
CNN-CM	0.225	-	0.906	-	0.877	-	-	-
JULE	0.369	-	0.913	0.964	0.915	0.961	0.848	0.684
CCNN	0.375	-	0.876	-	0.916	-	-	-
FR-UCNN	**0.401**	**0.456**	**0.919**	**0.971**	**0.920**	**0.974**	**0.891**	**0.783**

Table 4 compares the NMI and ACC performances of the proposed clustering method with other clustering methods with the parameters $T = 0.05$ and $\varepsilon = 0.01$. We will do the analysis of the results in Table 4 of non-deep-learning based clustering algorithms; these are not applicable for large-scale image dataset like ILRCVRC-12 in ImageNet. In another image dataset such as MNIST and YTF, FCM performed better than K-Mean. When we do the comparison with the results of deep-learning based clustering algorithms with the state-of-the-art clustering algorithms, at that point all deep-learning based clustering algorithms outperform with a significant margin. The reason behind this result that learning performs better is that feature representation of input images lead to better clustering results. We also compare our proposed clustering method with other deep-learning based clustering methods. The proposed FRUCNN achieves comparable performance upgrading in NMI by 0.021–0.032, 0.002–0.013, 0.004–0.005 and 0.043–0.089 with DEPICT, JULE and CCNN methods and significantly outperforms upgrading in NMI by 0.176–0.246. 0.013–0.103, 0.043–0.093

and 0.445 with DEC and CNN-CM for all ILSVRC 12-Val, MNIST, and YTF datasets. The experiment results demonstrate that the proposed FRUCNN approach performs better for image dataset for numerous scales.

Table 5 compares the NMI and ACC performance on different schemes of the proposed method. When we compare Schemes 1 and 2, the performance of clustering is improved with updating the centroid with the assistance of the FRCM algorithm. When Scheme 2 is compared with Scheme 1, it handles the large-scale image data with less time complexity because of the reliable sample selection for updating cluster centers by FRCM. The performance of Scheme 3, our proposed method, significantly improves the cluster performance in NMI with 0.201, 0.289, and 0361, when compared with only updating iteratively representation learning and in NMI with 0.191, 0.79, 0.068, and 0.151, when compared the proposed method is updating only cluster centroid. In the analysis from the above results, we achieved the better cluster performance with updating the cluster centers compared to updating iteratively representation learning. The highest performance of Scheme 3 shows that the combination of updating cluster centers and iterative representation learning simultaneously make a great combination for achieving maximum clustering performance.

Table 5. Comparison between different Schemes of Proposed Method Based on Performance Measure (NMI & ACC).

Methods	ILSVRC12-Val		MNIST-Full		MNIST-Test		YTF	
	NMI	**ACC**	**NMI**	**ACC**	**NMI**	**ACC**	**NMI**	**ACC**
Scheme–1	0.200	0.256	0.630	0.651	0.640	0.663	0.530	0.500
Scheme–2	0.310	0.357	0.840	0.852	0.852	0.860	0.740	0.700
Scheme–3	**0.401**	**0.456**	**0.919**	**0.971**	**0.920**	**0.974**	**0.891**	**0.783**

3.5.1. Computational Time Comparison

To evaluate the efficiency of our clustering algorithm on image datasets, we compare the computational time of our proposed FR-UCNN method with other compared algorithms, where we set the number of epochs for parameters updating to 10. Table 6 illustrates the computational time for FR-UCNN and other methods on all datasets. The comparison shows that our method consumed 3.1 h, 1.2 h, 0.16 h, and 0.5 h to obtain the clustering results of t ILSVRC12-Val, MNIST-Full, and MNIST-Test respectively. Our method achieved better results compared to other methods except CCNN due to a mini-batch based method with feature drift comparison which can effectively address the problem of large-scale image dataset.

Table 6. Computational Time Comparison of FRUCNN with compared methods.

Methods/Data	ILSVRC12-Val	MNIST-Full	MNIST-Test	YTF
DEC	-	-	-	0.4 h
DEPICT	-	2.7 h	0.41 h	0.6 h
CNN-CM	3.2 h	2.9 h	0.33 h	0.9 h
JULE	5.1 h	3.1 h	0.40 h	1.1 h
CCNN	1.2 h	-	-	-
FR-UCNN	3.1 h	1.2 h	0.16 h	0.5 h

3.5.2. Performance on Number of Cluster (k)

Figure 4 shows the NMI performance on the different number of k clusters w.r.t ILSVRC12-Val and MNIST-Full dataset. First, the ILSVRC12-Val dataset at k = 1 accomplishes the best NMI performance. The performance is decreasing with the number of k increasing. Let k = 1 be the good choice for ILSVRC12-Val, however the computation time is too long for fine-tuning the learning representation. We will consider the computational time and performance results together, at that

point the best choice is k = 10 or less for the ILSVRC12-Val dataset. Secondly, for the MNIST-Full dataset, at k = 9 accomplishes the highest NMI performance as expected. When we compare the performance MNIST-Full dataset with k = 9 and k = 10 then k = 9 accomplishes better performance as the handwriting digits 4 and 9 consider the same cluster. For both datasets, k = 9 or 10 seems to be a reasonable choice for good NMI performance with much less computation.

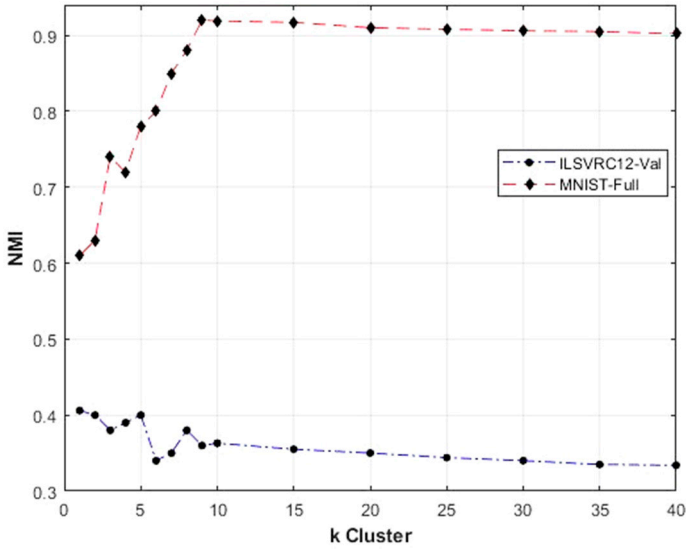
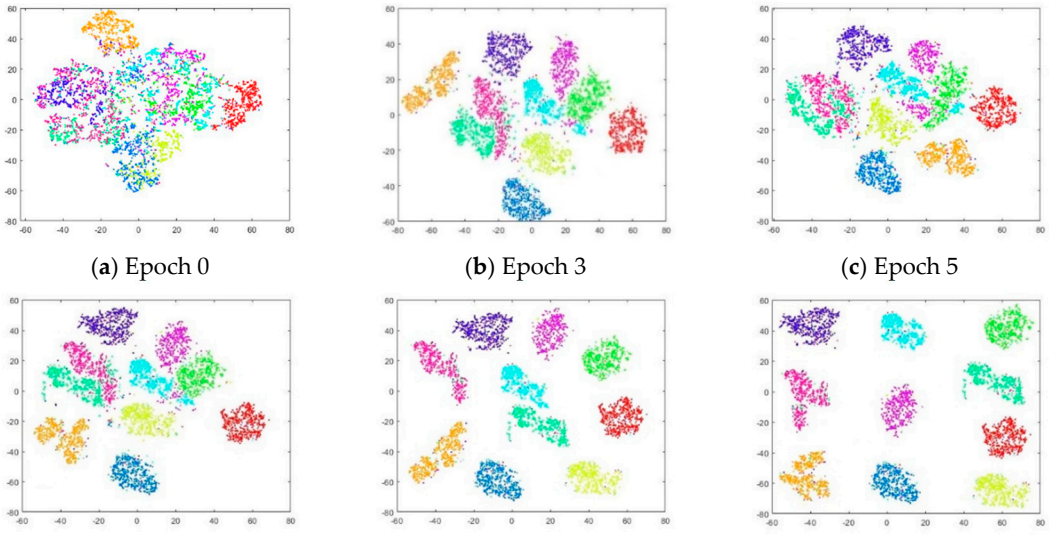


Figure 4. Comparison of NMI performance of the proposed method versus the number of k clusters.

3.5.3. Performance on Number of Epochs

Figure 5 demonstrates the importance of our proposed approach by comparing the different number of epochs on MNIST-train dataset using (t-SNE) [85] visualization, with true cluster labels shown in different colors. It can be seen from visualizations that the clusters are becoming increasingly well separated.



(**a**) Epoch 0 (**b**) Epoch 3 (**c**) Epoch 5

(**d**) Epoch 7 (**e**) Epoch 9 (**f**) Epoch 12

Figure 5. t-SNE (T-distributed Stochastic Neighbor Embedding) visualizations for clustering to show the discriminative capability of proposed approach on MNIST dataset. Note separation of clusters from epoch 0 to epoch 12.

Figure 6 also shows the comparison of NMI performance with respect to the different number of epochs on four data sets. We evaluate the performance of our proposed method; the clustering performance gradually increases with the number of epochs increasing for updating parameters. We also analyze from Figure 6 that the performance is stable after the number of epochs is 25 for ILSVRS12-Val and MNIST dataset and clustering performance is stable for YTF dataset after the number of epochs is 30. After the analysis of Figures 4 and 6, the best choice for ILSVRC12 is $k = 10$ with epochs 25 and for MNIST and YTF $k = 9$ or 10 with epochs 30.

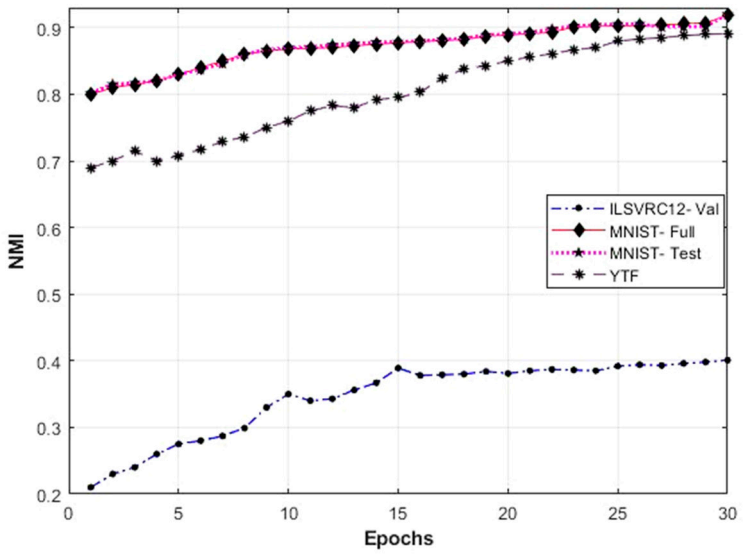


Figure 6. Comparison of NMI performances of the proposed method versus the number of training epochs for three datasets (ILSVRC12-Val, MNIST, and YTF).

4. Threats to Validity

Some potential threats to validity exist in our experimental study. Dataset quality might be the most important threat to the external validity, which refers to the generalizability of our experimental results. To guarantee the representativeness of our experiment, we utilized ImageNet, MNIST and YTF image dataset *s*, which are usually utilized for clustering techniques.

Our unsupervised CNN clustering method adopts an initial pre-trained CNN model from ImageNet to predict cluster labels; it can generally lead to consistent convergence performance. In addition, Fuzzy Rough C-Mean clustering is utilized for updating cluster centroid with CNN architecture. Although Fuzzy C-Mean clustering is certain to converge, there is no hypothetical guarantee for the convergence of the FRCM clustering approach.

We implement a broadly utilized metric NMI (Normalized Mutual Information) matrix and ACC (Average) to evaluate the clustering performance. NMI is more reliable for clustering results compared with other performance measures. To avoid the internal threat, all implementation is cross-checked by our research group. There is no hypothetical guarantee of convergence (just like the other existing deep-learning-based clustering models) for large image datasets. Our experiment results show that our approach can achieve a good convergence performance.

5. Conclusions

In this paper, we provided image clustering using extracted from convolutional clustering layers in a Convolution Neural Network (CNN). This provides a state-of-the-art performance; we have shown that with our proposed unsupervised CNN clustering based on Fuzzy Rough C-Mean (FRCM) algorithm, performance can be improved for robust large-scale image dataset based on the iteration between an updating cluster centroid using FRCM algorithm and an unsupervised CNN clustering

fine-tuning. An unsupervised CNN clustering can extract silent features from the convolutional clustering layer to produce the initial cluster center. During the training process, the cluster and representation are trained jointly; the cluster center is updated step-by-step by FRCM algorithm during the forward pass and learned representation in backward pass. We also show that reliable sample selection for updating cluster centroids by the FRCM algorithm is the key component to its success. Empirical studies with other non-fuzzy CNN reveal that fuzzy and rough learning with CNN demonstrates the strength of the proposed method on several image datasets. But the defect is that too many parameters need to adjusted, and in future we need to work to make the parameters self-adaptive.

Author Contributions: S.R. Conceptualization and Methodology; A.A. Software, Writing—review & editing; and L.J. Supervision and funding acquisition.

Acknowledgments: This work was supported in part by the National Basic Research Program (973 Program) of China (No. 2013CB329402), the National Natural Science Foundation of China (No. 61573267, 61473215, 61571342, 61572383, 61501353, 61502369, 61271302, 61272282, 61202176), the Fund for Foreign Scholars in University Research and Teaching Programs (the 111 Project) (No. B07048), the Major Research Plan of the National Natural Science Foundation of China (No. 91438201 and 91438103).

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Doersch, C.; Gupta, A.; Efros, A.A. Mid-level visual element discovery as discriminative mode seeking. In Proceedings of the Conference on Information Processing Systems, Lake Tahoe, NV, USA, 3–6 December 2013; pp. 494–502.
2. Han, D.; Kim, J. Unsupervised simultaneous orthogonal basis clustering feature selection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA, 7–12 June 2015; pp. 5016–5023.
3. Hariharan, B.; Malik, J.; Ramanan, D. Discriminative decorrelation for clustering and classification. In Proceedings of the European Conference on Computer Vision, Florence, Italy, 7–13 October 2012; pp. 459–472.
4. Nie, F.; Zeng, Z.; Tsang, I.W.; Xu, D.; Zhang, C. Spectral embedded clustering: A framework for in-sample and out-of-sample spectral clustering. *IEEE Trans. Neural Netw.* **2011**, *22*, 1796–1808. [PubMed]
5. Nie, F.; Dong, X.; Li, X. Initialization independent clustering with actively self-training method. *IEEE Trans. Syst. Man Cybern. Syst.* **2012**, *42*, 17–27. [CrossRef] [PubMed]
6. Song, J.; Gao, L.; Nie, F.; Shen, H.T.; Yan, Y.; Sebe, N. Optimized graph learning using partial tags and multiple features for image and video annotation. *IEEE Trans. Image Process.* **2016**, *25*, 4999–5011. [CrossRef] [PubMed]
7. Nie, F.; Wang, X.; Huang, H. Clustering and projected clustering with adaptive neighbors. In Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, New York, NY, USA, 24–27 August 2014; pp. 977–986.
8. Nie, F.; Wang, H.; Deng, C.; Gao, X.; Li, X.; Huang, H. New l1-Norm relaxations and optimizations for graph clustering. In Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, Phoenix, AZ, USA, 12–17 February 2016; pp. 1962–1968.
9. Gao, L.; Song, J.; Nie, F.; Zou, F.; Sebe, N.; Shen, H.T. Graph-without-cut: An ideal graph learning for image segmentation. In Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, Phoenix, AZ, USA, 12–17 February 2016; pp. 1188–1194.
10. Nie, F.; Ding, C.; Luo, D.; Huang, H. Improved minmax cut graph clustering with nonnegative relaxation. In Proceedings of the 2010 European Conference on Machine Learning and Knowledge Discovery in Databases: Part II, Barcelona, Spain, 20–24 September 2010; pp. 451–466.
11. Tian, F.; Gao, B.; Cui, Q.; Chen, E.; Liu, T.-Y. Learning deep representations for graph clustering. In Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence, Quebec, QC, Cannada, 27–31 July 2014; pp. 1293–1299.

12. Trigeorgis, G.; Bousmalis, K.; Zafeiriou, S.; Schuller, B. A deep semi-NMF model for learning hidden representations. In Proceedings of the 31st International Conference on International Conference on Machine Learning, Beijing, China, 21–26 June 2014; pp. 1692–1700.
13. Xie, P.; Xing, E. Integrating image clustering and codebook learning. In Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, Austin, TX, USA, 25–30 January 2015.
14. Guo, Y.; Ma, L.; Zhu, F.; Liu, F. Selecting Training Samples from Large-Scale Remote-Sensing Samples Using an Active Learning Algorithm. In *Computational Intelligence and Intelligent Systems*; Li, K.S., Li, J., Liu, Y., Castiglione, A., Eds.; Springer: Singapore, 2015; pp. 40–51.
15. Song, J.; Yang, Y.; Huang, Z.; Shen, H.T.; Luo, J. Effective multiple feature hashing for large-scale near-duplicate video retrieval. *IEEE Trans. Multimed.* **2013**, *15*, 1997–2008. [CrossRef]
16. Blum, A.L.; Langley, P. Selection of relevant features and examples in machine learning. *Artif. Intell.* **1997**, *97*, 245–271. [CrossRef]
17. Aggarwal, C.C.; Reddy, C.K. *Data Clustering: Algorithms and Applications*, 1st ed.; CRC Press: Boca Raton, FL, USA, 2013.
18. Gdalyahu, Y.; Weinshall, D.; Werman, M. Self-organization in vision: Stochastic clustering for image segmentation, perceptual grouping, and image database organization. *IEEE Trans. Pattern Anal. Mach. Intell.* **2001**, *23*, 1053–1074. [CrossRef]
19. Gowda, K.C.; Krishna, G. Agglomerative clustering using the concept of mutual nearest neighbourhood. *Pattern Recognit.* **1978**, *10*, 105–112. [CrossRef]
20. Herrero, J.; Valencia, A.; Dopazo, J. A Hierarchical Unsupervised Growing Neural Network for Clustering Gene Expression Patterns. *Bioinformatics* **2001**, *17*, 126–136. [CrossRef] [PubMed]
21. Ng, A.Y.; Jordan, M.I.; Weiss, Y. On spectral clustering: Analysis and an algorithm. In Proceedings of the 14th International Conference on Neural Information Processing Systems, Vancouver, BC, Canada, 3–8 December 2001; Volume 2, pp. 849–856.
22. Heyer, L.J.; Kruglyak, S.; Yooseph, S. Exploring Expression Data: Identification and Analysis of Coexpressed Genes. *Genome Res.* **1999**, *9*, 1106–1115. [CrossRef] [PubMed]
23. Ben-Dor, A.; Shamir, R.; Yakhini, Z. Clustering Gene Expression Patterns. *J. Comput. Biol.* **1999**, *6*, 281–297. [CrossRef] [PubMed]
24. Hartuv, E.; Shamir, R. A Clustering Algorithm Based on Graph Connectivity. *Inf. Process. Lett.* **2000**, *76*, 175–181. [CrossRef]
25. Jiang, D.; Pei, J.; Zhang, A. DHC: A Density-Based Hierarchical Clustering Method for Time-Series Gene Expression Data. In Proceedings of the Third IEEE Symposium on Bioinformatics and Bioengineering, Bethesda, MD, USA, 12 March 2003; pp. 393–400.
26. Yao, J.; Vasilakos, A.V.; Pedrycz, W. Granular computing: Perspectives and challenges. *IEEE Trans. Cybern.* **2013**, *43*, 1977–1989. [CrossRef] [PubMed]
27. Peters, G.; Crespo, F.; Lingras, P.; Weber, R. Soft clustering fuzzy and rough approaches and their extensions and derivatives. *Int. J. Approx. Reason.* **2013**, *54*, 307–322. [CrossRef]
28. Zhang, Y.; Ye, S.; Ding, W. Based on rough set and fuzzy clustering of MRI brain segmentation. *Int. J. Biomath.* **2017**, *10*, 1750026. [CrossRef]
29. Zhang, T.F.; Ma, F.M. Improved rough k-means clustering algorithm based on weighted distance measure with gaussian function. *Int. J. Comput. Math.* **2017**, *94*, 663–675. [CrossRef]
30. Vidhya, K.A.; Geetha, T.V. Rough set theory for document clustering: A review. *J. Intell. Fuzzy Syst.* **2017**, *32*, 2165–2185. [CrossRef]
31. Suri, N.N.R.R.; Murty, M.N.; Athithan, G. Detecting outliers in categorical data through rough clustering. *Nat. Comput.* **2016**, *15*, 385–394. [CrossRef]
32. Ye, M.; Liu, W.; Wei, J.; Hu, X. Fuzzy c-means and cluster ensemble with random projection for big data clustering. *Math. Probl. Eng.* **2016**, *2016*, 6529794. [CrossRef]
33. Yu, H.; Zhang, C.; Wang, G. A tree-based incremental overlapping clustering method using the three-way decision theory. *Knowl.-Based Syst.* **2016**, *91*, 189–203. [CrossRef]
34. Qian, P.; Sun, S.; Jiang, Y.; Su, K.-H.; Ni, T.; Wang, S.; Muzic, R.F., Jr. Cross-domain, soft-partition clustering with diversity measure and knowledge reference. *Pattern Recognit.* **2016**, *50*, 155–177. [CrossRef] [PubMed]
35. Barnett, V.; Lewis, T. *Outliers in Statistical Data*, 3rd ed.; John Wiley & Sons: New York, NY, USA, 1984.

36. Dunn, J.C. Some recent investigations of a new fuzzy partition algorithm and its application to pattern classification problems. *J. Cybern.* **1974**, *4*, 1–15. [CrossRef]
37. Lingras, P.; West, C. Interval set clustering of Web users with rough k-means. *J. Intell. Inf. Syst.* **2004**, *23*, 5–16. [CrossRef]
38. Peters, G. Outliers in rough k-means clustering. In Proceedings of the 1st International Conference on Pattern Recognition and Machine Intelligence, Kolkata, India, 20–22 December 2005; Springer: Berlin/Heidelberg, Germany; Volume 3776, pp. 702–707.
39. Jin, W.; Tung, A.K.H.; Han, J. Mining top-n local outliers in large databases. In Proceedings of the 17th ACMSIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, 26–29 August 2001; ACM Press: New York, NY, USA, 2001; pp. 293–298.
40. Krizhevsky, A.; Sutskever, I.; Hinton, G. ImageNet classification with deep convolutional neural networks. In Proceedings of the 25th International Conference on Neural Information Processing Systems, Lake Tahoe, NV, USA, 3–6 December 2012; pp. 1097–1105.
41. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
42. Simonyan, K.; Zisserman, A. Very deep convolutional networks for large-scale image recognition. In Proceedings of the International Conference on Learning Representations, San Diego, CA, USA, 7–8 May 2015.
43. Long, J.; Shelhamer, E.; Darrell, T. Fully convolutional networks for semantic segmentation. In Proceedings of the IEEE Conference Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 3431–3440.
44. Hsu, C.C.; Lin, C.W. CNN-Based Joint Clustering and Representation Learning with Feature Drift Compensation for Large-Scale Image Data. *IEEE Trans. Multimed.* **2018**, *20*, 421–429. [CrossRef]
45. Xie, J.; Girshick, R.; Farhadi, A. Unsupervised deep embedding for clustering analysis. In Proceedings of the 33rd International Conference on International Conference on Machine Learning, New York, NY, USA, 19–24 June 2016.
46. Dundar, A.; Jin, J.; Culurciello, E. Convolutional clustering for unsupervised learning. In Proceedings of the International Conference on Learning Representations, San Juan, PR, USA, 2–4 May 2016.
47. Yang, J.; Parikh, D.; Batra, D. Joint unsupervised learning of deep representations and image clusters. In Proceedings of the IEEE IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 5147–5156.
48. Yang, Q.; Wang, H.; Li, T.; Yang, Y. Deep Belief Networks Oriented Clustering. In Proceedings of the 2015 10th International Conference on Intelligent Systems and Knowledge Engineering (ISKE), Taipei, Taiwan, 24–27 November 2015; pp. 58–65.
49. Pados, D.A.; Papantoni-Kazakos, P. A note on the estimation of the generalization error and the prevention of overfitting [machine learning]. In Proceedings of the 1994 IEEE International Conference on Neural Networks (ICNN'94), Orlando, FL, USA, 28 June–2 July 1994; pp. 571–586.
50. Dietterich, T. Overfitting and Undercomputing in Machine Learning. *ACM Comput. Surv.* **1995**, *27*, 326–327. [CrossRef]
51. Chen, D.; Lv, J.; Yi, Z. Unsupervised multi-manifold clustering by learning deep representation. In Proceedings of the Workshops at the AAAI Conference on Artificial Intelligence, San Francisco, CA, USA, 4–5 February 2017; pp. 385–391.
52. Chen, G. Deep learning with nonparametric clustering. *arXiv*, 2015; arXiv:1501.03084.
53. Dizaji, K.G.; Herandi, A.; Huang, H. Deep clustering via joint convolutional autoencoder embedding and relative entropy minimization. *arXiv*, 2017; arXiv:1704.06327.
54. Harchaoui, W.; Mattei, P.-A.; Bouveyron, C. Deep adversarial Gaussian mixture autoencoder for clustering. In Proceedings of the International Conference on Learning Representations (ICLR), Toulon, France, 24–26 April 2017.
55. Hsu, C.-C.; Lin, C.-W. Cnn-based joint clustering and representation learning with feature drift compensation for large-scale image data. *arXiv*, 2017; arXiv:1705.07091.
56. Hu, W.; Miyato, T.; Tokui, S.; Matsumoto, E.; Sugiyama, M. Learning discrete representations via information maximizing self augmented training. *arXiv*, 2017; arXiv:1702.08720.

57. Huang, P.; Huang, Y.; Wang, W.; Wang, L. Deep embedding network for clustering. In Proceedings of the International Conference on Pattern Recognition (ICPR), Stockholm, Sweden, 24–28 August 2014; pp. 1532–1537.
58. Li, F.; Qiao, H.; Zhang, B.; Xi, X. Discriminatively boosted image clustering with fully convolutional auto-encoders. *arXiv*, 2017; arXiv:1703.07980.
59. Liu, H.; Shao, M.; Li, S.; Fu, Y. Infinite ensemble for image clustering. In Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, 13–17 August 2016; pp. 1745–1754.
60. Lukic, Y.; Vogt, C.; Dürr, O.; Stadelmann, T. Speaker identification and clustering using convolutional neural networks. In Proceedings of the 2016 IEEE 26th International Workshop on Machine Learning for Signal Processing (MLSP), Vietri sul Mare, Italy, 13–16 September 2016; pp. 1–6.
61. Premachandran, V.; Yuille, A.L. Unsupervised learning using generative adversarial training and clustering. In Proceedings of the International Conference on Learning Representations (ICLR), San Juan, PR, USA, 2–4 May 2016.
62. Saito, S.; Tan, R.T. Neural clustering: Concatenating layers for better projections. In Proceedings of the Workshop Track of International Conference on Learning Representations (ICLR), Toulon, France, 24–26 April 2017.
63. Wang, Z.; Chang, S.; Zhou, J.; Wang, M.; Huang, T.S. Learning a task-specific deep architecture for clustering. In Proceedings of the SIAM International Conference on Data Mining (ICDM), Barcelona, Spain, 12–15 December 2016; pp. 369–377.
64. Yang, B.; Fu, X.; Sidiropoulos, N.D.; Hong, M. Towards k-means-friendly spaces: Simultaneous deep learning and clustering. *arXiv*, 2016; arXiv:1610.04794.
65. Zheng, Y.; Tan, H.; Tang, B.; Zhou, H. Variational deep embedding: A generative approach to clustering. *arXiv*, 2016; arXiv:1611.05148.
66. Kappeler, A.; Morris, R.D.; Kamat, A.R.; Rasiwasia, N.; Aggarval, G. Combining deep learning and unsupervised clustering to improve scene recognition performance. In Proceedings of the 2015 IEEE 17th International Workshop on Multimedia Signal Processing (MMSP), Xiamen, China, 19–21 October 2015; pp. 1–6.
67. Kulkarni, M.; Karande, S.S.; Lodha, S. Unsupervised Word Clustering Using Deep Features. In Proceedings of the 2016 12th IAPR Workshop on Document Analysis Systems (DAS), Santorini, Greece, 11–14 April 2016; pp. 263–268.
68. Xu, J.; Xu, B.; Wang, P.; Zheng, S.; Tian, G.; Zhao, J.; Xu, B. Self-Taught convolutional neural networks for short text clustering. *Neural Netw.* **2017**, *88*, 22–31. [CrossRef] [PubMed]
69. Hu, Q.; Yu, D. An improved clustering algorithm for information granulation. In Proceedings of the 2nd International Conference on Fuzzy Systems and Knowledge Discovery (FSKD'05), Changsha, China, 27–29 August 2005; Springer: Berlin/Heidelberg, Germany, 2005; Volume 3613, pp. 494–504.
70. Bezdek, J.C. *Pattern Recognition with Fuzzy Objective Function Algorithms*; Kluwer Academic Publishers: New York, NY, USA, 1981.
71. Hsu, C.C.; Lin, C.W. Unsupervised convolutional neural networks for large-scale image clustering. In Proceedings of the 2017 IEEE International Conference on Image Processing (ICIP), Beijing, China, 17–20 September 2017; pp. 390–394.
72. Angiulli, F. Fast condensed nearest neighbor rule. In Proceedings of the ACM International Conference, Bonn, Germany, 7–11 August 2005; pp. 25–32.
73. Deng, J.; Berg, A.; Satheesh, S.; Su, H.; Khosla, A.; Li, F. ImageNet large-scale visual recognition competition 2012. *Int. J. Comput. Vis.* **2015**, *115*, 211–252. Available online: http://www.image-net.org/challenges/LSVRC/2012/ (accessed on 20 May 2018).
74. Oquab, M.; Bottou, L.; Laptev, I.; Sivic, J. Is object localization for free?—Weakly-supervised learning with convolutional neural networks. In Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA, 7–12 June 2015; pp. 685–694.
75. Avrithis, Y.; Kalantidis, Y.; Anagnostopoulos, E.; Emiris, I.Z. Web-scale image clustering revisited. In Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV), Santiago, Chile, 7–13 December 2015; pp. 1502–1510.

76. Nair, V.; Hinton, G.E. Rectified linear units improve restricted Boltzmann machines. In Proceedings of the 27th International Conference on International Conference on Machine Learning, Haifa, Israel, 21–24 June 2010; pp. 807–814.
77. The MathWorks, Inc. *Statistics Toolbox Release 2018a, MATLAB*; The MathWorks, Inc.: Natick, MA, USA, 2018.
78. LeCun, Y. The MNIST Database of Handwritten Digits. Available online: http://yann.lecun.com/exdb/mnist/ (accessed on 20 May 2018).
79. Wolf, L.; Hassner, T.; Maoz, I. Face recognition in unconstrained videos with matched background similarity. In Proceedings of the CVPR 2011, Colorado Springs, CO, USA, 20–25 June 2011; pp. 529–534.
80. Xu, W.; Liu, X.; Gong, Y. Document clustering based on non-negative matrix factorization. In Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Informaion Retrieval, Toronto, ON, Canada, 28 July–1 August 2003; pp. 267–273.
81. Huang, J.; Nie, F.; Huang, H.; Ding, C. Robust manifold nonnegative matrix factorization. *ACM Trans. Knowl. Discov. Data* **2014**, *8*, 11. [CrossRef]
82. Hand, D.J.; Till, R.J. A simple generalisation of the area under the ROC curve for multiple class classification problems. *Mach. Learn.* **2001**, *45*, 171–186. [CrossRef]
83. Anil, K.J. Data clustering: 50 years beyond K-Means. *Pattern Recognit. Lett.* **2010**, *31*, 651–666.
84. Li, K.; Cao, Z.; Cao, L.; Zhao, R. A novel semi-supervised fuzzy c-means clustering method. In Proceedings of the Chinese Control and Decision Conference, Guilin, China, 17–19 June 2009; pp. 3761–3765.
85. Van Der Maaten, L.; Hinton, G. Visualizing data using t-SNE. *J. Mach. Learn. Res.* **2008**, *9*, 2579–2605.