Improvement for Large-Scale Image Data using Fuzzy Rough C-Mean Based Unsupervised CNN Clustering: Case Study designbyhumans.com

Tuan Anh Tran1

First author's affiliation, an Institution with a very long name, xxxx@gmail.com

Quy Ban Tran1

Second author's affiliation, possibly the same institution, xxxx@gmail.com

Clustering analysis, specifically, for large image data are increasingly being applied in various fields such as finance, risk management, prediction, etc., and has been an interesting subject in many science discussions. Deep learning, a widely used approach, along with classical methods is being used to address sophisticated classification problems stem from real world cases. In this study we will be taking various approaches to the classification problems and, measure their effectiveness by combining different methods using the results taken from different scenarios in this study. Initial evaluations will be conducted on a dataset that we have been able to collect from the website designbyhumans.com and the result will be analyzed in detail with each technique and will be compared with other similar and different approaches.

1. Introduction

Since the beginning of the information technology era, image processing has become an important research field, its necessity has been heavily implied in computer vision applications, where even a single image can potentially contain valuable information. 3.2 billion images are uploaded to the Internet every day with different purposes, that huge number indicated the demand of integrating large scale image data into several different fields. A huge number of efforts have been made to meet those artificial demands, such as reducing a few samples from the already enlarged dataset or encoding image features within the dataset as there might and usually exists redundant and noise samples scattered throughout the dataset with varied magnitude.

Unsupervised learning is a popular machine learning tool to visualize the structure beneath the huge dataset. An unsupervised method must be able to reproduce the result without too much prior knowledge from previous learning from unlabeled datasets, which is commonplace in the real world and typically unknown before applying cluster analysis techniques. Distinctive or so-called “hard” clustering methods such as hierarchical, density-based, centroid-based and graph theoretical have been used extensively for prediction on these structures with commendable precision. However, with the existence of high-quality image data with great depth to their perception, these “hard” methods are left exposed to a huge challenge: The amount of immeasurable vagueness, uncertainty or overlapping of samples from the clusters is nearly impossible to make precise predictions to each sample.

Because of such uncertainties, Rough Set[1] and Fuzzy Set[2] theory have been introduced and implemented as cluster analysis methods, so-called “soft” methods opposed to “hard” methods in order to highlight the blurry nature of the data that might be an oversight to well-known “hard” methods. Two prominent methods are Fuzzy C-means[3] and Rough C-means[4], which use probability-based weighting values to identify the clusters. Rough C-means describes a cluster with a centroid which is initialized by choosing data points randomly, and a pair of lower and upper approximation with different weighting values. Fuzzy C-means on the other hand use weight calculation formula to impose different weighting values based on distances to cluster centers. Fuzzy Rough C-means (FRCM)[5] is the combination of these abovementioned to overcome both the overlapping nature and uncertainty of image data. It incorporated elements of the two: A center, a crisp lower approximation and a fuzzy boundary approximation, classifying based on rough approximation and calculating based on fuzzy approximation.

Deep learning is also a trending topic in recent years because of its specialization in solving certain problems, especially in analyzing large image data. Some architectures that represent its robustness include Deep Belief Network[6], Deep Boltzmann Machine[7], Deep Autoencoder[8] and Convolutional Neural Networks[9]. For video and imagery applications, we’ve already had CNN-based architectures: AlexNet[10], ResNet[11], VGG[12] and FCN[13]. These architectures demonstrated robust accuracy in image data classification tasks with sufficient labeled training set.

Clustering algorithm’s performance depends greatly on noise reduction and feature representation power. Deep learning architectures rely on accurate labeled data, which is usually not available in the real-world applications and does not applicable in cluster analysis applications. Therefore, the model can be pre-trained on existing dataset with reliable labeling process and using transfer learning and proper tuning to be able to identify such inconsistency.

Many efforts have been done to combine cluster analysis with representation learning. Hsu[14] proposed CNN-based joint clustering in which Mini-batch K-Means algorithm is executed to assign cluster labels. However, it carries the cons of the hard-clustering method, is the deterministic model that incomprehensible with the inconsistency of the image dataset.

In this paper, we will be using combined CNN and FRCM architecture provided from the paper[citation] but with some modification in order to fit in the basic concepts that has been cited in the given paper using our own dataset from the website designbyhumans.com and we will be comparing the performance between the dataset and MNIST dataset.

* 1. Background
     1. Fuzzy Rough C-means (FRCM) clustering algorithm:

As stated by Hu et al. [5], the FRCM algorithm is a combination of both Fuzzy and Rough C-means. FCM maps each data points into a membership matrix which ranges from 0 to 1, each point belongs to some or all the clusters to some degree probabilistically and calculate the centroids based on distances to each of the cluster centers that have been initialized by random sampling. RCM classifies the points into two parts: The lower approximation, upper approximation; those who belong in the lower approximation are guaranteed to be a data point of that cluster, and the ones in the upper part belong to a cluster by some extent with respect to their different weighting values. Inspired by these concepts, FRCM integrated all of these elements and imposed fuzzy membership values of each sample to the lower and upper area of the clusters.

Let a set of image data , where d is the dimension of the data points. Each cluster , where k denotes the number of clusters, is regarded as a rough set. The data points are categorized into the lower approximation and the upper approximation . Let be a vector composed of k centers of clusters, where . The points in lower approximation are guaranteed to be in the clusters and take the same weight value, while the ones that are in the upper region have diverse effects on the centers and clusters, therefore different weighting values must be imposed on only the points belong to the upper region to compute new cluster centers.

Let be a membership matrix, we have the membership function:

(1)

The exponent m > 1 is the parameter to change the weighting impact of membership values, usually in range from 1.5 to 2.5 as different methods suggested. The new cluster center is computed according to the function:

(2)

And the objective function for FRCM is as follows:

(3)

To wrap up, the algorithm is formulated as follows:

ALGORITHM 1: Fuzzy Rough C-Means

Unlabeled data I, number of clusters k, threshold parameter T, exponent index m, stop criterion .

Membership matrix u, k cluster centers

Let , initialize membership matrix u randomly

while , do

calculate pairwise distances between each data points and calculate cluster centers using randomly initialized membership matrix u and equation (2)

assign data points to the approximations:

calculate its closest center and distance set as follows:

classify data points based on distance set A:

compute new membership values using equation (1)

compute cost using equation (3)

* + 1. Fuzzy Rough UCNN Clustering Architecture:

To be able to enhance clustering algorithm’s performance, unsupervised CNN is integrated along with Fuzzy Rough C-Means clustering algorithm. There are two parts in this architecture: The pre-clustering stage and further joint clustering and representation learning. The clusters are updated by using the FRCM algorithm in the forward pass of the architecture and optimized by the stochastic gradient descent in the backward pass.

In the pre-clustering stage, the size of the dataset will determine how large the respective multi-convolutional layers will be and will generally requires large-scale networks in case the dataset is huge enough. For the ImageNet dataset used in this paper, the AlexNet architecture is proven to be effective enough to implement the training. The architecture consists of 5 convolutional layers (Conv1 – Conv5) used in AlexNet, followed along by 3 adjustment layers (Conv6, Conv7, CConv) with channel number 6144, 2048 and k, connected with a fully connected (FC) layer and a softmax layer to extract the image features and predict the cluster labels.

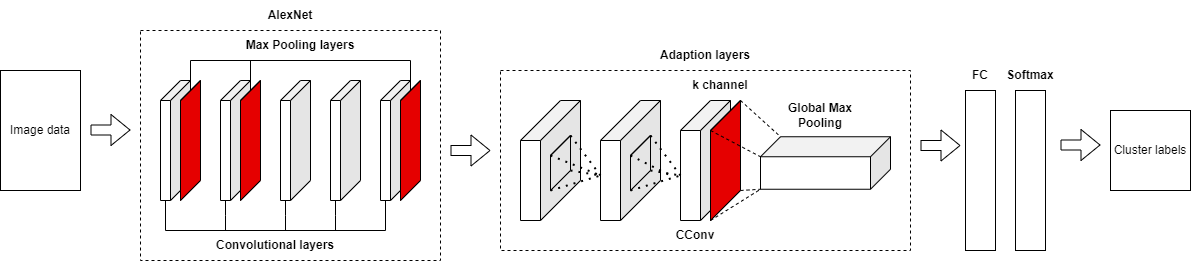


Figure 1: The FRUCNN architecture

* 1. Related Work

Hu et al. and Yu et al.[5] proposed a clustering algorithm which is the combination of Fuzzy C-means and Rough C-means that perceive real world applications of unstable and blurry data. Each data point is categorized based on threshold value and its membership value into lower and upper approximation. The centers are re-calculated using randomly created membership matrix and new membership value, correspond with data point’s coordinates and membership value, along with an exponent index value, usually stable at 2.0.

Hsu et al. and Lin et al.[14] proposed a clustering CNN architecture and representation learning method to cluster image data based on their extracted visual representations. A clustering algorithm is utilized to support the CCNN by assigning the labels to each image in the dataset as truth labels and cluster the extracted salient features at each selected output. The CCNN is the primary method as it uses randomly selected samples to feed forward and updates the extracted features of those samples and fine tune its parameters.

Riaz et al., Arshad et al., Jiao et al.[15] utilized a CCNN architecture based upon FRCM for large image data problem. The architecture has been proven to be better than existing methods for image data clustering problem

1. 2. Materials and Methods
   1. Image data pre-processing

To increase sample variety during the pre-clustering process, the data is pre-processed by using random flip horizontally and random crop at 227 by 227 pixels, as standards before deploying AlexNet to train Conv1 – Conv5 parameters.

* 1. Cluster centroids calculating

Let be the set of images. We will be using the membership matrix from the result of the ground truth from applying FRCM to the dataset with the extracted centroid features using UCNN from the FC layer as extracted dataset features initial centroids in order to minimize the gap between using independent features and dataset. The FRCM is performed to update the cluster centroids by the objective function:

And the centroids are calculated by the function:

* 1. Representation learning

By using the UCNN architecture, features of the images in the dataset are extracted by using the FC layer and salient features are extracted using CConv layer. To learn the parameter of FC and softmax layer, we will be using SGD process, where omega subscript r i end subscript belongs to the FC layer and omega subscript i j end subscript belongs to the softmax layer, respectively. We will start with the objective function:

where k is denoted as the number of clusters, y with overparenthesis on top subscript j is the predicted jth cluster label using the UCNN and y subscript j is the predicted jth cluster label using FRCM as pseudo ground truth. Then by using chain rule and calculating gradient, we have the functions for updating FC and softmax layers:

(5)

(6)

* 1. The complete algorithm:

ALGORITHM 2: Cluster centroids updating

Image dataset I, k number of clusters, randomly selected data , learning rate , max iteration , Extracted image features of centroids of FC layer.

Final cluster centroids, final weights ()

For t = 1 to , do:

Calculate cluster label using FRCM as ground truth on randomly utilized centroids to speedup algorithm’s convergence

Forward feed the centroids and extract the features from UCNN’s clustering layer and FC layer.

Update cluster centroids by using FRCM on randomly sampled centroids in the FC layer and find the predicted cluster label based on the updated centroids.

Update the weights of FC layer and softmax layer by using function (5) and (6)

Fine tune the UCNN by using the objective function

end

* 1. Experiment:
     1. Data preparations:

In this paper, Python 3.9 was utilized as our programming tool because of its compatibility with various libraries and frameworks. ILSVRC19 train set in ImageNet is used in pre-training stage with our representation of AlexNet; it consists of over 1.2 million training images of size 256 x 256-pixels collected from 1000 categories. As mentioned above, our provided by DesignedByHumans dataset is used to evaluate the performance of this approach, which consists of over 52000 images of size 1200 x 1200-pixels downscaled to 28 x 28-pixels collected from 69 different categories includes sub-categories and possibly belong to many categories as once, which emulates real world example of unstable data for semi-supervised learning method.

The hardware we will be using is a personal computer using Ryzen 7 4800HS CPU and NVIDIA GTX 1650 GPU with 16GB of RAM updated with newest drivers on Tensorflow 2.9.

Table 1: Datasets used for evaluation

| Dataset | No. of samples | No. of classes | Image size |
| --- | --- | --- | --- |
| ILSVRC19 | 1,281,167 | 1000 | 256x256 |
| DesignByHumans dataset | 52,361 | 69 | 28x28 |

* + 1. Evaluation metrics:

Since we will be using our dataset, as stated in data preparation step, our dataset has a possibility of different samples belong to many categories, sub-categories included, overlapping each other on different clusters, so the metrics we will be using are popular metrics used to evaluate unsupervised learning methods with unknown ground truth.

The Silhouette Coefficient score, which is used when the ground truth is unknown and evaluation is performed using the model. A higher silhouette coefficient score indicates a model with better defined clusters:

where a is the mean distance between a sample and all other points in the same class, b is the mean distance between a sample and all other points in the next nearest cluster. The best value is 1 and the worst is -1 and the value near 0 indicates overlapping clusters.

The Calinski-Harabasz score is the ratio of the sum of between-clusters dispersion and of within dispersion of all clusters, for a set of data E of size which has been clustered into k clusters:

where is trace of the between group dispersion matrix and is the trace of the within-cluster dispersion matrix defined by:

with is the set of points in cluster q, is the center of cluster q, is the center of E, and is the number of points in cluster q

The Davies-Boulding score signifies the average similarity between clusters:

where is constructed by using:

with is the average distance between each point of cluster i and the centroid of that cluster and is the distance between cluster centroids i and j

* + 1. Comparison and evaluate:

To evaluate the performance of the method and compare it with other methods using the same dataset, we will be conducting our tests in comparison with KMeans since 1. KMeans is the most popular clustering algorithm used in most of the comparison and 2. It’s performance in synthetic and large data is still questionable in terms of overall performance

* + 1. Implementations and practices:

We began using the pre-trained AlexNet on the updated ILSVRC19 training set of ImageNet as our basic convolutional model and data augmentation methods: random flip left and right as well as random cropping samples to size 227x227 during the pre-training process to increase samples variety.

The proposed model consists of 5 convolutional layers taken from the pre-trained AlexNet, with the same configurations as the original model, concatenated with 2 adjustment layers with filter size 6144 & 2048, with kernel size 3 x 3 and a clustering convolutional layer with k channel size and the same kernel size as the adjustment layers, and followed by a global max pooling layer with 1 x k output, k is denoted as the number of clusters.

We evaluated the model on our dataset, consists of over 50000 RGB images of size 1200 x 1200 . The output of the FC layer is considered the centroids of the clusters and is updated in the forward propagation stage using FRCM and SGD in the backward propagation stage.

Our dataset is resized to 28 x 28 due to GPU memory constraints, as well as the model, shown in figure 2

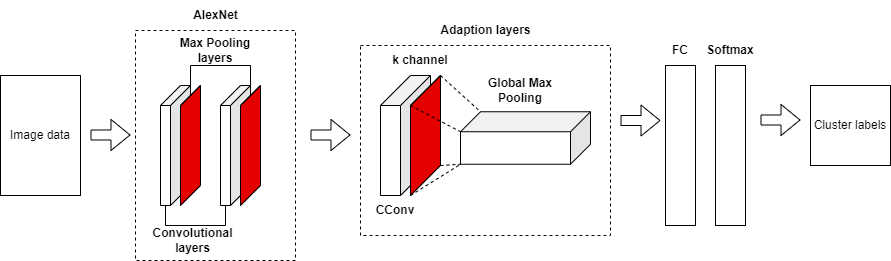


Figure 2: The implemented FRUCNN model

1. Results
   1. Performance measures:

Figure 2 shows the elbow graph which is used to estimate the number k clusters in DesignByHumans dataset. Since the dataset is not properly labeled and may belong to multiple categories, we will be using unsupervised method of estimating the optimal number of clusters instead of rely on other methods such as a normalized mutual information graph.

As seen in figure 3, k = 12 is the optimal choice for cluster number considering computational time and performance on the DesignByHumans dataset using the model. KMeans estimation also shows k = 12 is the optimal choice for our unlabeled dataset.

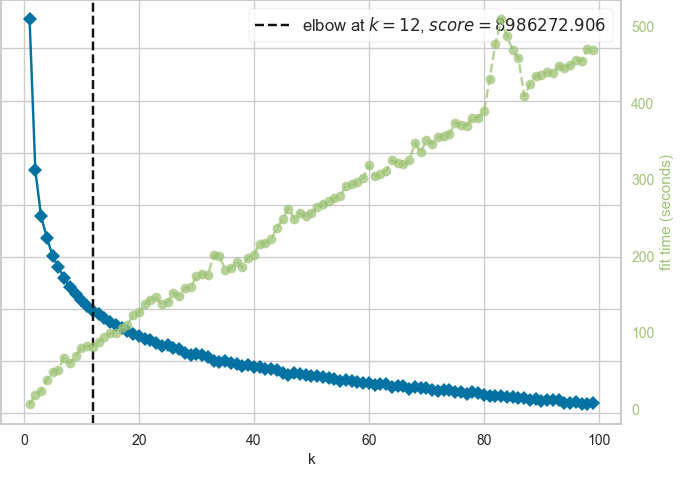


Figure 3: Elbow method graph for estimating the optimal number of clusters k in DesignByHumans dataset.

Table 2 compares the FRUCNN method with KMeans on our DesignByHumans dataset using Silhouette Coefficient, Calinski-Harabasz Index and Davies-Bouldin Index for comparison metrics with parameters T = 40 and = 0.01

Table 2: Results of performance measures (Silhouette Coefficient, Calinski-Harabasz Index, Davies-Boulding Index) of FRUCNN in comparison with KMeans on DesignByHumans dataset

| Methods | Silhouette Coefficient | Calinski-Harabasz Index | Davies-Bouldin Index |
| --- | --- | --- | --- |
| FRUCNN | 0.330 | 0.838 | 41957.614 |
| KMeans | 0.203 | 0.981 | 29106.019 |

* 1. Computational time:

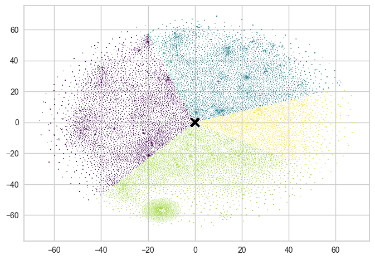
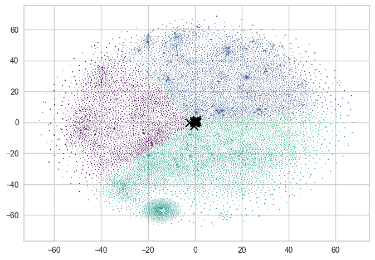
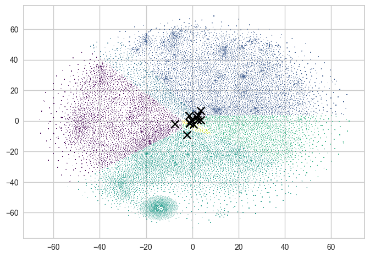
Another evaluation of the FRUCNN is the computational time, which demonstrates how efficient the method is in a period of time. The FRUCNN architecture took a much bigger time for the process to converge because the calculation is more complex than that of KMeans; the approximations took much longer to estimate than the usual hard clustering methods.

Table 3: Computational Time of FRUCNN in comparison with KMeans by hours on DesignByHumans dataset

| Methods | Time |
| --- | --- |
| FRUCNN | 6.3 h |
| KMeans | 1.89 h |

* 1. Impact on performance:

Figure 4 demonstrates the performance impact of the FRUCNN approach on different epochs on the DesignByHumans dataset, with cluster labels are referred by different colors. The visualization of the dataset shows that clusters were increasingly separated when the epochs increase. This is an important factor along with choosing the optimal parameters for such large dataset.

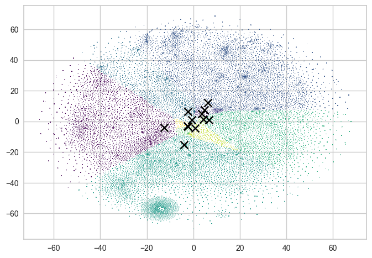
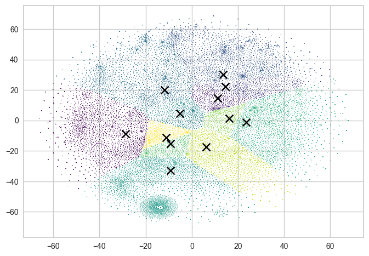
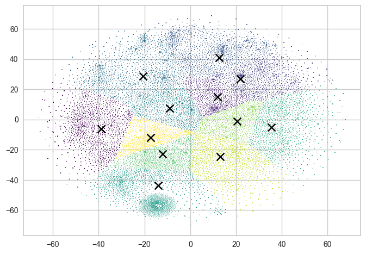
  

Figure 4: t-SNE visualizations for clustering to show the performance impact on different number of epochs on DesignByHumans dataset. Note the cluster centroids concentration and re-distribution of clusters from epoch 0 to 12

(From left to right, up and down: Epoch 0, 3, 5, 6, 9 and 12)

Figure 5 shows the Silhouette Coefficient performance on different number of epochs. The Silhouette Coefficient were increasing with increasing number of epochs which indicates that the more number of epochs, the better separation between different clusters. The silhouette coefficient decreases until the number of epochs reached 3, and increase until the number of epochs reached 73. After that point, the silhouette coefficient had minimal increases even when the number of epochs reached 70. So the optimal number of epochs in this case will be 70.

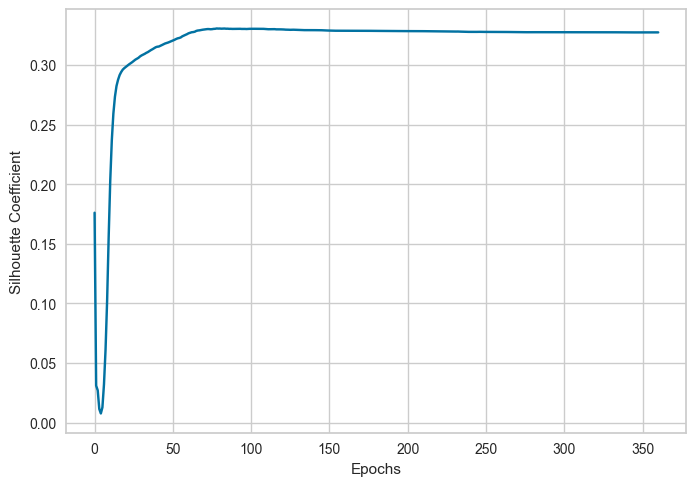


Figure 5: Silhouette Coefficient correspondence with different number of epochs

1. Discussion

Dataset quality is our main concern because as stated, the number of features in a HD quality image is demanding for the hardware, and the fact that we had to face constraints in computational power reflected this. The model is also taxing on the GPU memory that the performance may suffered heavily.as

FRCM is also an inconsistent method, as it tends not to converge in many situations during our implementation. Although the Fuzzy C-means is guaranteed to reach a convergence point, there is no hypothetical guarantee that FRCM algorithm reach convergence for large and inconsistent dataset as ours

1. Conclusions

In this documentation, we have applied the CNN model on our synthetic dataset based on the Fuzzy Rough C-means clustering algorithm. The algorithm can provide and be improved for better results on large scale, high dimensions image data by using iterations between updating cluster centroids using FRCM algorithm and fine-tuning process of the initial model. The CNN model was able to effectively extract salient features from the clustering layer as cluster centroids and features and updated in the forward pass. Comparison with KMeans proved the effectiveness of the method on selective dataset and the robustness of the algorithm with one of the most popular unsupervised learning methods. However, its uncertainty and defect in the operation will be obstacles that need to be address and improved upon. Being self-adaptive will be the objective of this learning method in the future

REFERENCES

1. P. Zdzisław, "Rough set theory and its applications", Journal of Telecommunications and Information Technology, vol. 3, pp. 7-10, 2002.
2. Zimmermann, H.-J. (2010), Fuzzy set theory. WIREs Comp Stat, 2: 317-332. <https://doi.org/10.1002/wics.82>
3. James C. Bezdek, Robert Ehrlich, William Full, FCM: The fuzzy c-means clustering algorithm,Computers & Geosciences,Volume 10, Issues 2–3, 1984,Pages 191-203,ISSN 0098-3004,https://doi.org/10.1016/0098-3004(84)90020-7.
4. Ubukata, S., Notsu, A. and Honda, K., 2017. General formulation of rough C-means clustering. International Journal of Computer Science and Network Security, 17(9), pp.29-38.
5. H. Qinghua and Y. Daren, "An Improved Clustering Algorithm for Information Granulation", 2005.
6. Hinton, G.E., 2009. Deep belief networks. Scholarpedia, 4(5), p.5947.
7. Salakhutdinov, R. and Larochelle, H., 2010, March. Efficient learning of deep Boltzmann machines. In Proceedings of the thirteenth international conference on artificial intelligence and statistics (pp. 693-700). JMLR Workshop and Conference Proceedings.
8. Zhou, C. and Paffenroth, R.C., 2017, August. Anomaly detection with robust deep autoencoders. In Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining (pp. 665-674).
9. O'Shea, K. and Nash, R., 2015. An introduction to convolutional neural networks. arXiv preprint arXiv:1511.08458.
10. Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. 2012. ImageNet classification with deep convolutional neural networks. In Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1 (NIPS'12). Curran Associates Inc., Red Hook, NY, USA, 1097–1105.
11. He, K., Zhang, X., Ren, S. and Sun, J., 2016. Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 770-778).
12. Simonyan, K. and Zisserman, A., 2014. Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556.
13. Long, J., Shelhamer, E. and Darrell, T., 2015. Fully convolutional networks for semantic segmentation. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 3431-3440).
14. Hsu, C.C. and Lin, C.W., 2017. Cnn-based joint clustering and representation learning with feature drift compensation for large-scale image data. IEEE Transactions on Multimedia, 20(2), pp.421-429.
15. Riaz, S., Arshad, A. and Jiao, L., 2018. Fuzzy rough C-mean based unsupervised CNN clustering for large-scale image data. Applied Sciences, 8(10), p.1869.