# Image Classification Using Network Pruning and Clustering Techniques

Ashok Chakravarthy Nara

The Australian National University, Acton ACT 2601, AU. n.ashokchakravarthy.1995@gmail.com

**Abstract.** Image Classification can be the concerned and difficult problem in the Image processing. Dealing of categorization and classification problems became comfortable after the invention of different neural network algorithms. In this paper , I propose different Image classification techniques using the progressive Image compression method, Fuzzy C means clustering (FCM) algorithm and Fuzzy C means (FCM) combined with Genetic Algorithm(GA) . For the accuracy comparison , I have implemented a simple baseline neural network. In the proposed image compression method, The compression is done in the form of pruning units which are not previously pruned in a tensor by zeroing out the ones with the lowest L1-norm and it is unstructured pruning which operates on weights of neurons.The FCM is soft clustering algorithm which membership to each data points rather than hard clustering which gives cluster value. The GA is selected for finding optimal features that are more relevant to the target. The data-set used for the whole experiment is Vehicle-X synthetically generated Image data in the form of features. The Data-set is divided into three categories randomly for experimenting in the form of Training , Validation and test. The accuracy results have been different for all proposed methods starting with simple neural network baseline model achieved is 4.11% continuing with progressive image compression method is 5.39% , then FCM gave 10.75% and finally FCM combined with GA gave 12.70%

**Keywords:** Image Compression · Pruning · Image Classification · L1-norm · Fuzzy c means Clustering Algorithm (FCM) · Genetic Algorithm.

## 1 Introduction

Image Classification is considered to be typical in the Digital image analysis and processing. Based on the new design of neural network algorithms in the past decades, Most of the typical problems involving classification and categorization are given the solution. In this paper, different image classification techniques are discussed- Progressive Image compression technique by network pruning [1], Fuzzy c means (FCM) clustering [7] and GA [8] combined with FCM .

### 1.1 Need for Network Pruning

Most of the Neural network Algorithms are expensive in terms of computational time , memory storage and needing the complex hardware. The Network pruning is the method of reducing size of the network and improve the generalization. The redundant neurons are removed based on the weights but the respective neurons to be removed is decided by the different aspects like angle between the activation vector [1] , lowest L1-norm . etc.

**Structured vs Unstructured approach** : According to [5], [6], the structured pruning is removing the layer or network's largest part where Unstructured pruning is done based on less important connections.

*In this paper, The Image compression method is done using unstructured approach where neurons having least L1-norm value were removed. If the structured approach is applied the whole layer or largest part of the network was to be removed which can be wrong way or mislead the technique.*

### 1.2 Need for FCM Clustering

The clustering is a process of exploring patterns or structures in a data-set. The clustering is an unsupervised learning which doesn't require labels while learning. Objects in each cluster are similar and objects in different clusters are dissimilar [7].

**Hard vs Soft Clustering** : In Hard Clustering, Each data point can belong to only one cluster and the membership is given in binary which is either 0 or 1. The down side of hard clustering is points near to cluster edge or near to other cluster may not be correctly classified.In other words, the edge points are not more when compared to points at center of cluster. In other case which is soft clustering, A data point can belong two or more clusters and membership is given in values 0 to 1. In soft clustering , the disadvantage of hard clustering is solved with membership value, if the membership is more for a cluster then the data points belongs to particular cluster.

*FCM is a soft clustering where membership is updated after each iteration depending on fuzziness parameter which should be greater than 1 or else it would be K-nearest neighbour algorithm.In this paper, FCM clustering was being used because of data-points are closely packed and hard clustering could be bad option.*

### 1.3 Why Genetic Algorithm?

The number of features in the data-set and large artificial neural network model makes the computational time expensive and also could negatively effect the performance. One possible solution is optimise the features which contributes more to the target variables [8].

*The main motivation of approaches in image classification - (Network pruning and Genetic Algorithm with FCM clustering) is achieving the desired goal at a loss or removal of some information during the process which happens in real world scenarios which is like having minimal information, can we classify or categorize the huge data.*

## 2 Data-set and Analysis

The data-set used for the experiments is Vehicle-X which is generated synthetically. The data is provided with features extracted from Resnet which is pretrained on ImageNet [2] .The file name to each file is given in the format "00001_c001_33.npy" where "00001" is the vehicle Id(class label) , "c001" is the camera Id and "33" is the counting number. The size of the each feature array is "2048 X 1".

Overall the dataset contains 77516 images of 1362 vehicles (classes) generated by different aspects consideration like Camera aspects ( Id's , Distance , height) , Light aspects (Direction, Intensity) , colors and Orientation of vehicle. The Vehicle-x Data-set is divided into three folders like Training , Validation and Test randomly. After Observing (Fig. 1) The vehicle id "1" has the more frequency among all the labels . Though the given feature array for a image has the values in range between 0 and 1 , the normalisation has done on the data frame after loading.

There are 11 types of vehicles in the data-set and the names of the vehicle types are in order with type labels 0 to 10 is Sedan, Suv, Van, Hatchback, Mpv, Pickup, Bus, Truck, Estate, Sportscar, RV. From (Fig.1), it is observed that the vehicle with type id "0"(sedan) has the maximum frequency in the dataset.

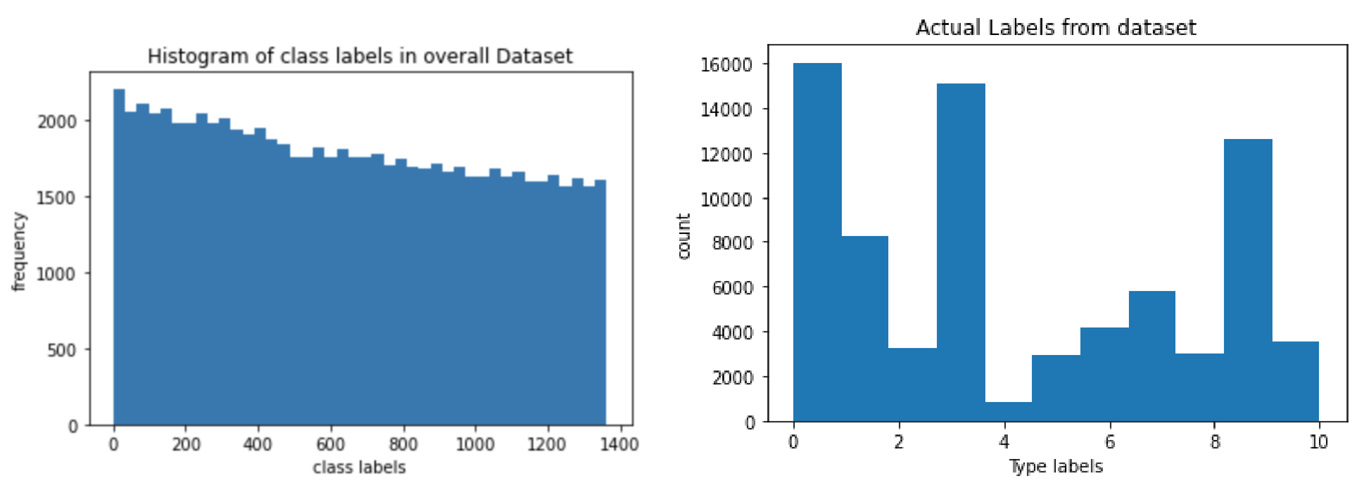


**Fig. 1.** Histogram of Vehicle-X Data-set

### 2.1 Network Architecture

The Architecture for the whole method of network pruning method remains the same in terms of number of layers and their size respectively.But when compared to Base-line model , the neuron count will be reduced in the pruned network. Starting from left most layer is the input layer (which is of size of input features) then middle layer is hidden layer (here taken size is half of input layer) and right most layer is output layer or targets (size of targets ). So the hyper parameters are input layer is "2048", hidden layer is "1024" and output layer is "1362"

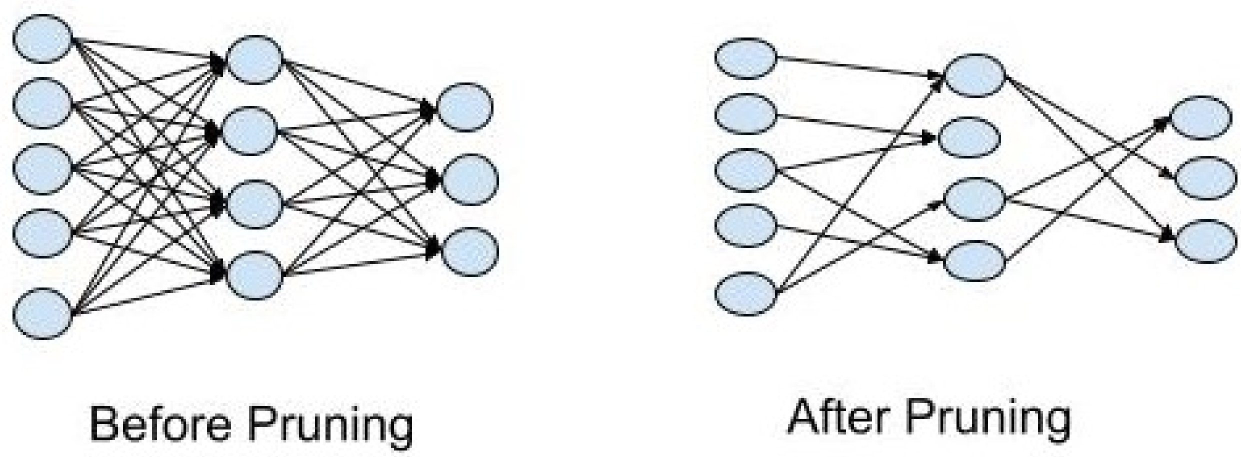


**Fig. 2.** Network Architecture of the pruning process

## 3 Implementation

### 3.1 Network Pruning

First the simple baseline model is implemented with one hidden layer of size equal to the half of the input features. The input layer and output layer has the size 2048 (size of feature array) and 1362 ( number of classes ) respectively. The given data is in the form of numpy array of size 2048 each. A data-frame of size of folder X (features size +1 ) is created and converted to CSV file then, feed to the network. The Last column is class labels and remaining data is normalised which is followed for all data-set folders.

According to [ [3], [4]]The pruning method is unstructured and neurons are removed on weight parameter on the hidden layer. The inbuilt functions "prune.global_unstructured()" are used with amount of neurons differing. The optimizer in both cases is Adam and loss function is cross entropy loss.

The parameters for the function are parameters to prune , pruning method and amount of neurons after each epoch. Here , I have used weight parameters, L1Unstructured method and amount ranges from 0.2 to 0.5. The pruning happens in the form of neurons having least L1-norm value will be removed and percentage of neurons to be removed are decided by amount.Due to the Computational cost and requirement of upscale hardware , The code was written using torch libraries and executed on Google scholar colaboratory. The Hyper Parameters chosen for Network Pruning method is shown in Table 1. The number of epochs 100 is decided because as the data-set is huge and increasing the epochs count would be over-fitting where increase in accuracy is shown in Table 2 from 500 to 100.

**Table 1.** Chosen Hyper Parameters for the Network Pruning Method

| Hyper Parameters | Values |
|---|---|
| input size | 2048 |
| hidden size | 1024 |
| number of classes | 1362 |
| number of epochs | 100 |
| batch size | 100 |
| learning rate | 0.01 |
| amount for pruning | 0.2 to 0.5 |

### 3.2 Fuzzy c means clustering Algorithm

For FCM clustering algorithm implementation, same data frame is used which is of same size but in place of class labels , type labels are appended and passed to the model. The parameters for the FCM are number of clusters , Maximum Iterations to which the FCM has to run and update the membership matrix and finally "m" value which is fuzziness value which should be always 1 or else it would become K Nearest Neighbouring Algorithm. According to [7],The sequence steps for FCM clustering is as follows:

1. Randomly initialize the cluster membership values, $\mu_{ij}$
2. calculating cluster centers

$$c_j = \frac{\sum_{i=1}^{N} u_{ij}^m * X_i}{\sum_{i=1}^{N} u_{ij}^m} \tag{1}$$

3. updating the membership value $\mu_{ij}$ according to the following

$$u_{ij} = \frac{1}{\sum_{k=1}^{c} \left( \frac{\|X_i - c_j\|}{\|X_i - c_k\|} \right)^{\frac{2}{m-1}}} \tag{2}$$

4. Getting the clusters or calculate the objective function $J_m$

Continue the steps 2-4 until maximum iterations limit is reached or $J_m$ improves by a less than threshold and finally calculate the partition matrix.According to [9], the conditions for calculating the parttion matrix is

$$\mu_{ik} \in [0,1], 1 \le i \le c, 1 \le k \le N \tag{3}$$

$$\sum_{i=1}^{c} \mu_{ik} = 1, 1 \le k \le N \tag{4}$$

$$0 < \sum_{i=1}^{c} \mu_{ik} < N, 1 \le i \le c \tag{5}$$

The FCM clustering algorithm's objective function $J_m$ is as follows

$$J_m = \sum_{i-1}^{N} \sum_{j-1}^{C} u_{ij}^m \| X_i - c_j \|^2 \tag{6}$$

- m is the fuzziness parameter which is any real number greater than 1
- N is the number of data
- c is the number of clusters
- $u_{ij}$ is the degree of membership of $x_i$ in cluster j
- $x_i$ is the $i^{th}$ of d-dimensional measured data
- $c_j$ is the d-dimension centre
- $\|\|$ is any norm expressing the similarity between any measured data and the centre.

### 3.3 Genetic Algorithm

Genetic Algorithm is used to calculate the optimal features which are more relevant to the target variables.According to [8],The Algorithm followed for optimal selection of features is as follows

1. Define chromosome Representation.
2. Initialize Population.
3. Evaluate fitness values for each chromosome.
4. Terminate solution.
   - If condition is satisfied then End the process.
   - Condition is not satisfied to step 5.
5. Apply Crossover to the population.
6. Apply Mutation to the population.
7. Evaluate the Fitness values for each new chromosome in the population.
8. Select chromosomes that will proceed to the next generation.

Continue the steps 5-9 until the condition is satisfied at step 5. Once the optimal feature list is obtained based on optimal silhoutte score [10]. When Observing the Silhoutte scores in Table 2 for FCM clustering algorithm is applied on training data-set, clears that clusters are assigned in wrong way. And the need for optimal features rises which contribute more to the target variables.

**Table 2.** Silhoutte score for different iterations on Training Dataset

| Iterations | Silhoutte Score |
|---|---|
| 5 | -0.02647 |
| 10 | -0.02632 |
| 15 | -0.01232 |

## 4 Experiments

### 4.1 Network Pruning

Once the data is converted to the required format multiple experiments were conducted based on epoch and amount combinations. With epoch count of 500 and learning rate of 0.01 , The baseline model gave 5.29% testing accuracy and with epoch count of 100 , the baseline model's test accuracy is 4.11%.

The pruning network model is experimented with different combinations of epoch count and amount. In all executions complete data-set folders where given to the models in the form of data-frames.

### 4.2 FCM and GA

The different combinations of fuzziness value "m" and iterations count are experimented which will be discussed in Results section. Due to computational time higher for GA execution only one combition is tried with parameters of population size and iterations are 10 and 2 respectively which gave accuracy of 12.73 %

## 5 Results and Discussion

### 5.1 Network Pruning and Simple Baseline model

The Table 3 shows the baseline model of two different epoch counts . Increase in the epoch count resulted in the accuracy. The Table 4 shows the accuracy of pruned model for different values . As the amount of neurons is increased , the accuracy reduced which makes the relation inversely related. The reason could be for reduced accuracy , after every epoch the L1-norm is calculated and the neurons having least l1-norm are removed depending on the amount provided.

As the pruning method is unstructured, the over fitting could be the problem and may lead to the accuracy reduction. From the above results , the proposing parameters are epochs are 100 and amount of neurons to be removed is "0.2". Since the model is implemented with using three layers , the parameters are proposed.But, the things gets change for the complicated models and simpler data of less features.

**Table 3.** Accuracy table for base-line model

| (epochs) | Training | Validation | Test |
|---|---|---|---|
| 500 | 95.30% | 5.32% | 5.29% |
| 100 | 78.60% | 4.16% | 4.11% |

**Table 4.** Accuracy table for pruned network model

| (epochs , amount) | Training | Validation | Test |
|---|---|---|---|
| (100, 0.2) | 76.66% | 5.12% | 5.39% |
| (100, 0.3) | 75.09% | 5.12% | 4.96% |
| (100, 0.4) | 73.97% | 4.81% | 4.76% |
| (100, 0.5) | 73.03% | 4.61% | 4.46% |

### 5.2 FCM and GA

The combinations of iterations and m are shown in Table 5 and 6. Due to computational cost is higher and data-set size is huge only few combinations are experimented and respective plots of histogram of type labels are shown in (Fig.3). Though iterations 15 showed higher frequency , the silhoutte score is negative and suffers from wrong allocation. Increasing the iteration count could lead to the over-fitting problem. For optimal value of fuzziness parameter "m" is 1 to 2 which gives maximum accuracy. The complete data-set is made into single data-frame with type labels as targets and fed into FCM. The parameters are iterations as 10, clusters as 11, m as 1.7 which gave accuracy as 10.23%

**Table 5.** Accuracy for different iterations on Training Data-set with m=1.7 and cluster=11

| Iterations | Accuracy |
|---|---|
| 5 | 12.63% |
| 10 | 10.75% |
| 15 | 15.63% |

## 6 Future Work

In Future, Different combinations of FCM parameters need to be investigated. Trying for betterment of accuracy is stopped because of over-fitting problem and computational time. Trying for optimal feature list using correlation between the features which can be one experimental cause for different technique. Different pruning techniques using other deep learning techniques like Conventional auto encoders and Knn algorithms. Need to investigate the structured way of pruning techniques for the image classification. If the network model is huge with large size which makes a possibility of structured approach of network pruning can give results.

**Table 6.** Accuracy for different iterations on Training Data-set with Iterations=5 and cluster=11

| m | Accuracy |
|---|---|
| 1.7 | 12.63% |
| 2.5 | 7.56% |
| 3.5 | 8.09% |
| 4.5 | 11.09% |
| 5.5 | 8.64% |

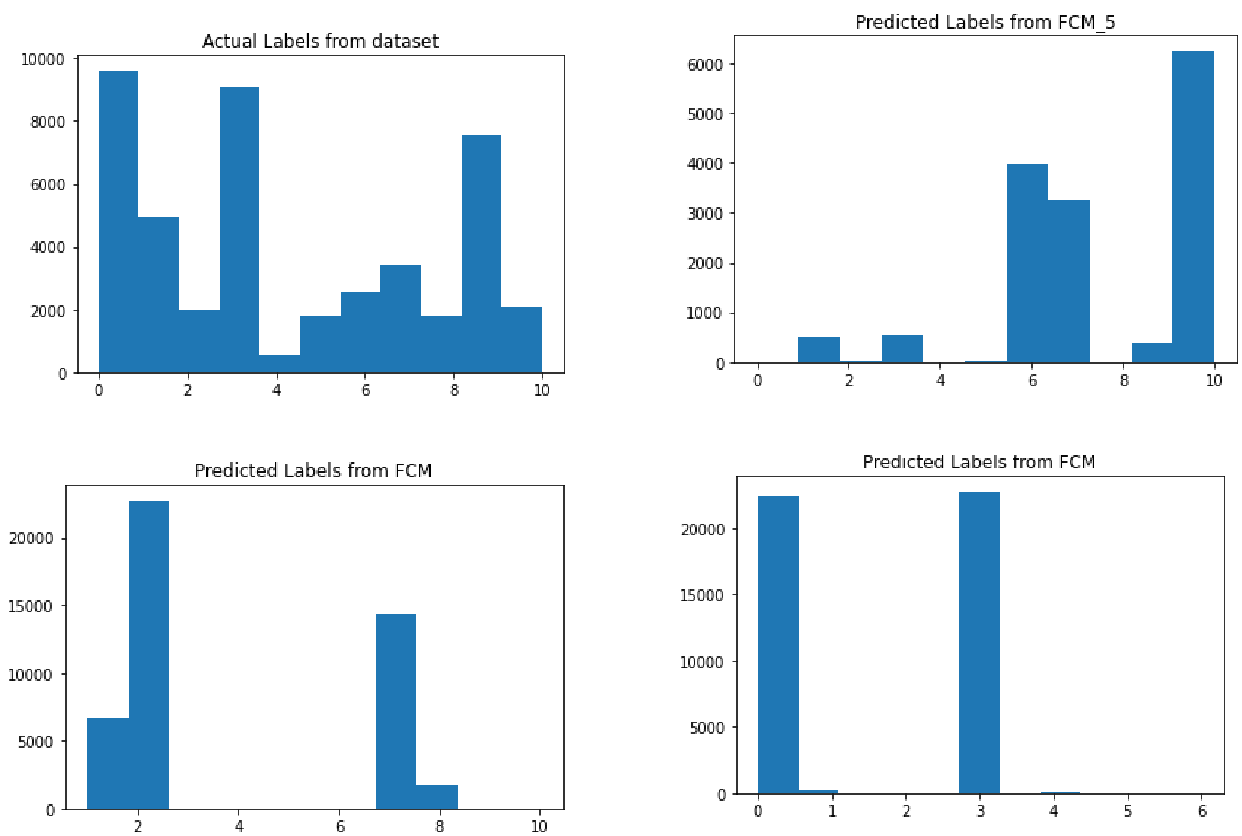


**Fig. 3.** Histogram of Actual Type labels and different iterations count. Top left is Actual type labels and top right was with iterations as 5 , bottom left was with iterations as 10 , bottom right was with iterations as 15

## References

1. Gedeon, T.D., Harris, D.: Progressive Image Compression. IJCNN. (1992)
2. Yao, Y., Zheng, L., Yang, X., Naphade, M., Gedeon T.D. : Simulating Content Consistent Vehicle Datasets with Attribute Descent. EECV, (2020).
3. PRUNING TUTORIAL, https:// pytorch.org/tutorials/intermediate/pruning_tutorial.html . Last accessed 24 April 2021
4. Paganini, M., Forde, J.: Streamlining Tensor And Network Pruning in PYTORCH. Workshop Paper at ICLR (2020)
5. Li, H., Kadav, A., Durdanovic, I., Samet, H., and Graf,H. P. Pruning filters for efficient convnets.
6. The Case for Sparsity in Neural Networks, Part 1: Pruning, https://numenta.com/blog/2019/08/30/case-for-sparsity-in-neural-networks-part-1-pruning. Last accessed 25 April 2021.
7. *Fuzzy Logic Fuzzy Clustering*,Lecture Notes, Neural Networks, Deep Learning and Bio-inspired Computing COMP8420, Australian National University, delivered 19 March 2021
8. Gedeon, T.D., Sharma, N : *Genetic Algorithm for Feature selection*,Lecture Notes, Neural Networks, Deep Learning and Bio-inspired Computing COMP8420, Australian National University, delivered 18 March 2021
9. Ruspini, E. H., Numerical methods for fuzzy clustering. Inform. Sci. 2, 319–350.(1970)
10. Silhoutte Coeficient, https://towardsdatascience.com/silhouette-coefficient-validating-clustering-techniques-e976bb81d10c. Last accessed 31 May 2021.