Clustering Image Data on their External Properties Using K-Means Clustering

Soumya Nasipuri
Machine Learning Intern
AI Technology and Systems
Kolkata, India
nasipurisoumya@gmail.com

***Abstract*— Image data has a various range of external properties like color, orientation shapes and size and in today's world image data is becoming more complex and vastly detailed. Also, in this era of Deep Learning and Machine Learning, Image Data clustering is a big necessity. Self-driving vehicles use image clustering and image segmentation during object detection phase. This paper mainly focuses on clustering fruits data with K-Means clustering and at first it uses feature extraction of the image data using a pretrained Convolutional Neural Network. It clusters the image data on the basis of their external properties.**

Keywords—Image data, Deep Learning, Machine Learning, Self-driving vehicles, Convolutional Neural Network, Object Detection, K-Means Clustering

I. INTRODUCTION

Image data or a digital image is a unique kind of data, image data consists of pixel values and in computers it is represented as an array, in the array the values of the pixel range from 0 to 255. Also, the image data has two formats as in coloured and grayscale data. If it is a coloured data, it has three channels as RGB – Red, Green, Blue. The pixel value on each pixel on each channel represents the amount of colour or pixel depth or no. of bits per pixel it has for that particular channel in particular pixel. If a single pixel of the red channel has a value of 134, it means that the pixel has a pixel depth or intensity of red of 134. This paper focuses on how the digital image is clustered on the basis of many external properties which is being recognised by the differences in the pixel depth on the images. The images are fed forward through the pretrained VGG–16[1] model. The output of the model will be the feature extracted from the image. Then, PCA [2] or principal component analysis is used to extract the two most significant features of the feature data. Then K-Means Clustering [3] is applied on the two most significant components of the image data. K-Means Clustering is unsupervised machine learning algorithm to clustered unlabelled data.

II. METHODOLGY

Methodology states that the methods which are used for the successful completion of the research. Methodology is the key ingredient for any kind of research as it determines the rationality and the optimization used to complete the research. Methodology gives us ample evidence of the fact that the certain research or paper is plagiarized.

A. Data Collection

The data which is used for this paper is the Fruits-360[4] dataset available in Kaggle. The dataset is basically a fruits dataset which has various types or classes of fruits and each fruit class contains numerous numbers of pictures taken by a Logitech C920 camera in various angles and orientation. The file format of the data is:

Filename format: image_index_100.jpg (e.g. 32_100.jpg) or r_image_index_100.jpg (e.g. r_32_100.jpg) or r2_image_index_100.jpg or r3_image_index_100.jpg. "r" stands for rotated fruit. "r2" means that the fruit was rotated around the 3rd axis. "100" comes from image size (100x100 pixels).

The other properties of the dataset are as follows:

Image size: 100 x100 pixels

Number of classes: 118 (fruits and vegetables).

Different varieties of the same fruit (apple for instance) are stored as belonging to different classes.

The classes of the dataset are Apples (different varieties: Crimson Snow, Golden, Golden-Red, Granny Smith, Pink Lady, Red, Red Delicious), Apricot, Avocado, Avocado ripe, Banana (Yellow, Red, Lady Finger), Beetroot Red, Blueberry, Cactus fruit, Cantaloupe (2 varieties), Carambula, Cherry (different varieties, Rainier), Cherry Wax (Yellow, Red, Black), Chestnut, Clementine, Cocos, Dates, Ginger Root, Granadilla, Grape (Blue, Pink, White (different varieties)), Grapefruit (Pink, White), Guava, Hazelnut, Huckleberry, Kiwi, Kaki, Kohlrabi, Kumsquats, Lemon (normal, Meyer), Lime, Lychee, Mandarine, Mango (Green, Red), Mangostan, Maracuja, Melon Piel de Sapo, Mulberry, Nectarine (Regular, Flat), Onion (Red, White), Orange, Papaya, Passion fruit, Peach (different varieties), Pepino, Pear (different varieties, Abate, Forelle, Kaiser, Monster, Red, Williams), Pepper (Red, Green, Yellow), Physalis (normal, with Husk), Pineapple (normal, Mini), Pitahaya Red, Plum (different varieties), Pomegranate, Pomelo Sweetie, Potato (Red, Sweet, White), Quince, Rambutan, Raspberry, Redcurrant, Salak, Strawberry (normal, Wedge), Tamarillo, Tangelo, Tomato (different varieties, Maroon, Cherry Red, Yellow), Walnut.

B. *Data Preprocessing*

The image file for each class are of size 100 x 100 pixels and the Convolutional model used in the feature extraction phase requires that the data should be of 224 by 224 pixels. So, the data should be pre-processed before feeding forward into the CNN model.

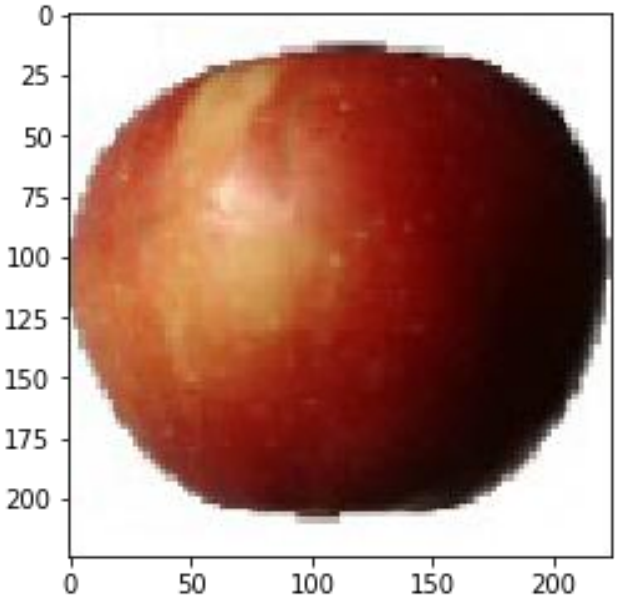

Figure 1: Image of Apple Braeburn(224 x 224 pixels)

Pixel Normalisation [5]: It is a normalisation technique where we rescale the pixel values. In image data pixel values range from 0 to 255 but due to the high values like this it is computationally very expensive. Due to this difficulty we reduce the range of the pixel depth significantly. The normalised pixel range is zero to one. The formulae used for this normalisation is:

$$I_N = (I - Min) * \left(\frac{(Max_N - Min_N)}{(Max - Min)}\right) + Min_N \quad (1)$$

Here, I = The original pixel value to be normalised

I_N =The new pixel value

Max = The old maximum pixel value = 255

Min = The old minimum pixel value = 0

Min_N = The new minimum pixel value = 0

Max_N = The new maximum pixel value = 255

By putting those values in (1), we get,

$$I_N = \frac{I}{255} \quad (2)$$

The next step in this data pre-processing section is reshaping the image array. The image array provided in the dataset of each is of shape 100 by 100 pixels. But in the VGG-16 model, the input image should have a image array of shape 224 by 224 pixels. So, in this step the image array is stretched to the required input size.

Now, the dataset is reduced because there are many classes of the same fruit class. This is very much redundant due to the fact that this paper focuses on external properties of image array which is shape, orientation and colour. The different species of the same class already have semi identical external features, so, if those are used in the feature extraction phase, the results will be biased and inaccurate. Also, using the vast data of the Fruits-360 will be computationally very expensive, reducing the data size will reduce the cost of computation. The information regarding the new dataset are as follows:

No. of classes: 57

Total number of trainable images: 27,342

C. *Feature Extraction Using Pre-trained Model*

For the feature extraction part, we need to use a method which can recognise patterns and can extract the main features of the images in the dataset. For that we are using a pretrained model VGG-16(Visual Geometry Group). VGG-Net consists of 16 convolutional layers and is very appealing because of its very uniform architecture. It is the runner-up on the ILSVRC 2014[6]. The model has a total of 16 layers which is why it is called the VGG-16 model. Also, this 16 layer is divided into 5 convolutional blocks and 1 Fully Connected block and followed by a classification activation layer (Softmax or Sigmoid Activation function), each convolutional block consists of multiple convolutional layers where each is followed by a ReLU(Rectified Linear Unit) activation layer. Also, each convolutional block has a max pooling layer at the end. Since, we are using the model for feature extraction phase we don't require the fully connected blocks and the classification activation layer. The model has a total of 14,714,688 trainable parameters which requires a massive amount of computation power to train. Due to this limitation, we used a pre-trained VGG-16 model. The model is trained with the ImageNet [7] Data. For the feature extraction we input the image in the model after pre-processing and expanding the dimension of the image so the first argument in the shape denotes the number of training examples. After that the model return the output which is of shape (m, 7, 7, 512) where m denotes the number of training examples. In this case m = 1 because we are inputting the images one by one in the model for extraction.

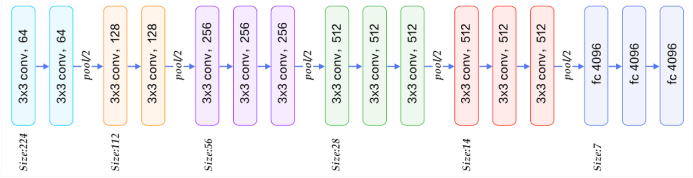

Figure 2: VGG-16 Model

D. *Principal Component Analysis(PCA)*

Principal Component Analysis, or PCA, is a dimensionality-reduction method that is often used to reduce the dimensionality of large data sets, by transforming a large set of variables into a smaller one that still contains most of the information in the large set.

Reducing the number of variables of a data set naturally comes at the expense of accuracy, but the trick in dimensionality reduction is to trade a little accuracy for simplicity. Because smaller data sets are easier to explore and visualize and make analyzing data much easier and faster for machine learning algorithms without extraneous variables to process. The algorithm [8] which is used for PCA is as follows:

- The procedure to implement PCA is very intuitive, the first step of the PCA is to standardize the values. The latter stages in the PCA requires variance which is very sensitive. The higher range variable will dominate the lower range variable e.g. the 1 – 100 range will dominate the 0 – 1 range variable. The mathematical formulae that is used in standardization is:

$$z = \frac{value - mean}{standard\ deviation} \quad (3)$$

- The next step is covariance matrix computation. The aim of this step is to understand how the variables of the input data set are varying from the mean with respect to each other, or in other words, to see if there is any relationship between them. Because sometimes, variables are highly correlated in such a way that they contain redundant information. So, in order to identify these correlations, we compute the covariance matrix.

$$\begin{bmatrix} \mathrm{Con}(x,x) & \mathrm{Con}(x,y) & \mathrm{Con}(x,z) \\ \mathrm{Con}(y,x) & \mathrm{Con}(y,y) & \mathrm{Con}(y,z) \\ \mathrm{Con}(z,x) & \mathrm{Con}(z,y) & \mathrm{Con}(z,z) \end{bmatrix} \quad (4)$$

  The value of Con(x,x) = Variable x and Con(x,y) = Con(y,x). So, only the values of the upper triangle are important to us. Now if the value is positive(+ve), it means that variables are correlated and they will decrease or increase together and if the value is negative(-ve) then the variables are not correlated.
- Eigenvectors and eigenvalues are the linear algebra concepts that we need to compute from the covariance matrix in order to determine the ***principal components*** of the data. This is the next step. Principal components are new variables that are constructed as linear combinations or mixtures of the initial variables. These combinations are done in such a way that the new variables (i.e., principal components) are uncorrelated and most of the information within the initial variables is squeezed or compressed into the first components. So, the idea is 10-dimensional data gives you 10 principal components, but PCA tries to put maximum possible information in the first component, then maximum remaining information in the second and so on, until having something like shown in the scree plot below.

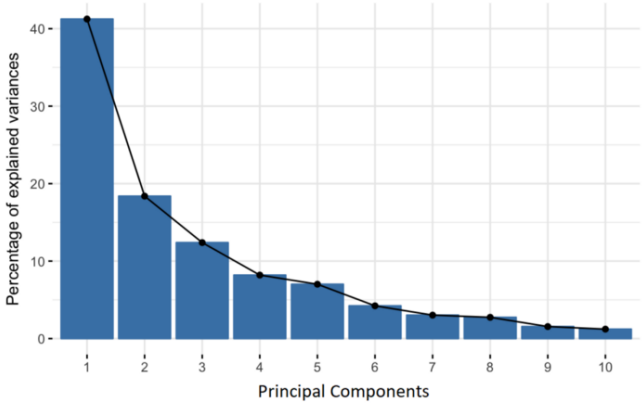

Figure 3: Percentage of variance (information) for by each PC

- In the next step, we are going to form the feature vector. As we saw in the previous step, computing the eigenvectors and ordering them by their eigenvalues in descending order, allow us to find the principal components in order of significance. In this step, what we do is, to choose whether to keep all these components or discard those of lesser significance (of low eigenvalues), and form with the remaining ones a matrix of vectors that we call *Feature vector*. This is the step where the dimensionality is reduced. If we choose to keep only p components out of n ($p < n$) then dimensions of the data are reduced.
- In the previous steps, apart from standardization, you do not make any changes on the data, you just select the principal components and form the feature vector, but the input data set remains always in terms of the original axes (i.e, in terms of the initial variables).
  In this step, which is the last one, the aim is to use the feature vector formed using the eigenvectors of the covariance matrix, to reorient the data from the original axes to the ones represented by the principal components (hence the name Principal Components Analysis). This can be done by multiplying the transpose of the original data set by the transpose of the feature vector.

$$Data_{Final} = FeatureVector^T * Data_{Original}{}^T \quad (5)$$

E. Image Clustering using K-Means Clustering

K-Means clustering is an unsupervised machine learning algorithm. In unsupervised learning, unlabeled data and it clusters the data which have a similar pattern or in other words, “Similar Data”. This algorithm works on the principle of centroids and Euclidian distance mainly. The *K*-means clustering algorithm uses iterative refinement to produce a final result. The algorithm inputs are the number of clusters *K* and the data set. The data set is a collection of features for each data point. The algorithms start with initial estimates for the *K* centroids, which can either be randomly generated or randomly selected from the data set. The algorithm [9] then iterates between two steps:

- **Data assignment step:** Each centroid defines one of the clusters. In this step, each data point is assigned to its nearest centroid, based on the squared Euclidean distance. More formally, if c_i is the collection of centroids in set C, then each data point x is assigned to a cluster based on

$$argmin_{c_i \in C} dist(c_i, x)^2 \qquad (6)$$

where *dist*() is the standard (L_2) Euclidean distance [10]. Let the set of data point assignments for each i^{th} cluster centroid be S_i.

- **Centroid update step:** In this step, the centroids are recomputed. This is done by taking the mean of all data points assigned to that centroid's cluster.

$$c_i = \frac{1}{|S_i|} \sum_{x_i \in S_i} x_i \qquad (7)$$

The algorithm iterates between steps one and two until a stopping criterion is met (i.e., no data points change clusters, the sum of the distances is minimized, or some maximum number of iterations is reached).
Disadvantages [11] of this algorithm are as follows:

- The learning algorithm requires apriority specification of the number of clusters centres.
- The use of Exclusive Assignment - If there are two highly overlapping data then k-means will not be able to resolve that there are two clusters.
- The learning algorithm is not invariant to non-linear transformations i.e. with different representation of data we get different results (data represented in form of cartesian co-ordinates and polar co-ordinates will give different results).
- Euclidean distance measures can unequally weight underlying factors.
- The learning algorithm provides the local optima of the squared error function.
- Randomly choosing of the cluster centre cannot lead us to the fruitful result.
- Applicable only when mean is defined i.e. fails for categorical data.
- Unable to handle noisy data and outliers.
- Algorithm fails for non-linear data set.

III. Results of the Clustering Algorithm

After completion of the project and applying the methods described earlier, we derived few observations and results for conclusion and evaluation. The conclusion and the evaluation are the most important part of any project or research. It determines the value and the importance of the given project or research.

A. Clustering of the Dataset

In the K-Means clustering, we used K = 10. We made a data-frame of the post PCA data. We used only two principal components for the data-frame. Then, we used K-Means clustering with K = 10 and trained the data-frame. After the training, we plot the data-frame with 10 clusters as different colours and black semi-transparent dots to represent the centroids of the ten clusters. The X-axis of the plot is the Principal component 1 with the higher variance and the Y-axis of the plot is the principal component 2 with the comparatively lower variance. The picture of the plot is given below:

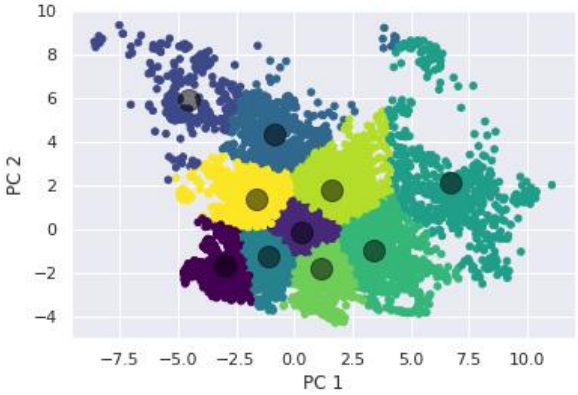

Figure 4: Plot of the Dataset Points After Clustering

B. Cluster Properties

The 10 cluster which are formed during the K-Means Clustering have similar properties. The clustering algorithm clusters the fruits on the basis of their various external properties. Here is an example of a cluster:

Figure 5: Cluster number 6

In this cluster we can observe that the there are three classes of fruits which are Pineapple Mini, Pithaya Red, Kohlrabi. We can observe that these fruits are the only fruits in the classes which have spikes on their outer shell. The cluster is formed on the basis of the spikes and unusual shape of the respective fruits on the dataset.
Here is an example of another cluster which have another similar external property in which they are grouped together:

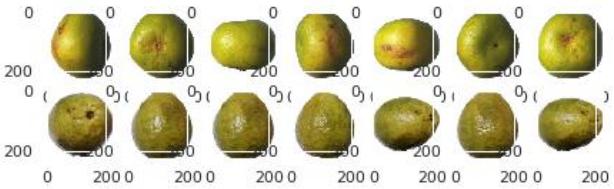

Figure 6: Cluster number 9

In this cluster, there are two classes of fruits which are Pomelo Sweetie and Guava. We can observe that these fruits are clustered or sampled out together because of their color and their shape. The classes or fruits both have approximately an oval like shape and the fruits have a greenish color which look very similar.

IV. Conclusion

At the conclusion of this paper we can say that unsupervised clustering algorithm especially, K-Means Clustering, groups the image data together on the basis of their external properties. We need a convolutional neural network which is pre-trained or it can be trained and then use it only to extract feature (we have to remove the classification layer) and then the features can be clustered together on the basis of their values. Here is another cluster which shows similar external property of shape and fading external colour (very light colours)

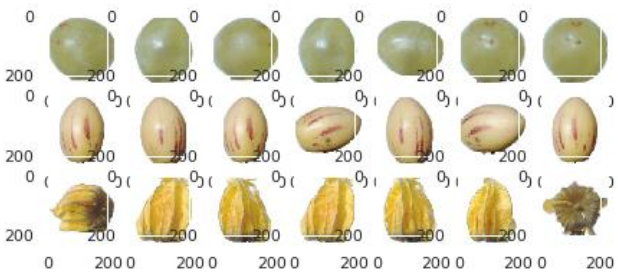

Figure 7: Cluster number 0

References

[1] arXiv:1409.1556 [cs.CV]

[2] 10.1002/wics.101

[3] Lloyd, Stuart P. "Least squares quantization in PCM." Information Theory, IEEE Transactions on 28.2 (1982): 129-137.

[4] Horea Muresan, Mihai Oltean, Fruit recognition from images using deep learning, Acta Univ. Sapientiae, Informatica Vol. 10, Issue 1, pp. 26-42, 2018.

[5] https://en.wikipedia.org/wiki/Normalization_(image_processing)

[6] Olga Russakovsky*, Jia Deng*, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg and Li Fei-Fei. (* = equal contribution) **ImageNet Large Scale Visual Recognition Challenge**. *IJCV*, 2015. paper | bibtex | paper content on arxi

[7] http://www.image-net.org/

[8] https://towardsdatascience.com/a-step-by-step-explanation-of-principal-component-analysis-b836fb9c97e2

[9] https://www.datascience.com/blog/k-means-clustering

[10] https://iq.opengenus.org/euclidean-distance/

[11] https://sites.google.com/site/dataclusteringalgorithms/k-means-clustering-algorithm