**Dictionary learning** is a branch of signal processing and machine learning that aims at finding a frame (called dictionary) in which some training data admits a sparse representation. The sparser the representation, the better the dictionary.  
Our project is Dictionary learning. We take Images of ‘N’ number of people, each persons having Ni number of images and we vectorise the images of each person as rows or columns . After the vectorization, we did K- Means clustering of all the vectors of Nth person and we got Means of these images and the labels of all these vectors were also stored. We did co-variance of these images having same label. Then we did M-Spherical means and we get the dictionary. We didn’t use any Features.  
  
INTRODUCTION :   
  
Sparse **dictionary learning** is a representation **learning** method which aims at finding a sparse representation of the input data (also known as sparse coding) in the form of a linear combination of basic elements as well as those basic elements themselves. These elements are called atoms and they compose a **dictionary**. Atoms in the dictionary are not required to be [orthogonal](https://en.wikipedia.org/wiki/Orthogonal_basis), and they may be an over-complete spanning set. This problem setup also allows the dimensionality of the signals being represented to be higher than the one of the signals being observed. The above two properties lead to having seemingly redundant atoms that allow multiple representations of the same signal but also provide an improvement in [sparsity](https://en.wikipedia.org/wiki/Sparsity" \o "Sparsity) and flexibility of the representation.

One of the key principles of dictionary learning is that the dictionary has to be inferred from the input data. The emergence of sparse dictionary learning methods was stimulated by the fact that in [signal processing](https://en.wikipedia.org/wiki/Signal_processing) one typically wants to represent the input data using as few components as possible. Before this approach the general practice was to use predefined dictionaries (such as [fourier](https://en.wikipedia.org/wiki/Fourier_transform" \o "Fourier transform) or [wavelet](https://en.wikipedia.org/wiki/Wavelet_transform) transforms). However, in certain cases a dictionary that is trained to fit the input data can significantly improve the sparsity, which has applications in data decomposition, compression and analysis and has been used in the fields of image [denoising](https://en.wikipedia.org/wiki/Noise_reduction" \o "Noise reduction) and [classification](https://en.wikipedia.org/wiki/Image_classification), video and [audio processing](https://en.wikipedia.org/wiki/Audio_signal_processing). Sparsity and overcomplete dictionaries have immense applications in image compression, image fusion and inpainting.  
  
If Dictionary contains more atoms than the dimension of the image signal then it is called as a ‘ Over Complete Dictionary ‘   
  
*K*-means clustering is a type of unsupervised learning, which is used when you have unlabeled data (i.e., data without defined categories or groups). The goal of this algorithm is to find groups in the data, with the number of groups represented by the variable *K*. The algorithm works iteratively to assign each data point to one of *K* groups based on the features that are provided. Data points are clustered based on feature similarity. The results of the *K*-means clustering algorithm are:

1. The centroids of the *K* clusters, which can be used to label new data
2. Labels for the training data (each data point is assigned to a single cluster)

Rather than defining groups before looking at the data, clustering allows you to find and analyze the groups that have formed organically. The "Choosing K" section below describes how the number of groups can be determined.

Each centroid of a cluster is a collection of feature values which define the resulting groups. Examining the centroid feature weights can be used to qualitatively interpret what kind of group each cluster represents.    
Spherical K-Means Clustering:

The Spherical K-means algorithm, i.e., the K-means algorithm with cosine similarity, is a popular method for clustering high-dimensional text data. In this algorithm, each vector as well as each cluster mean is represented as a high-dimensional unit-length vector. That is, each cluster mean vector is updated, only after all document vectors being assigned, as the (normalized) average of all the document vectors assigned to that cluster. It is very useful in clustering of large data sets. The basic methodology here is, firstly, we found the mean of whole data set and updated the whole data set by subtracting it from every vector. This differentiated each vector making closer ones go farther, and farther ones come. We then converted each data into a unit vector, assigned certain levels to each data set and found the mean vector of the respective levels. We later converted them into unit vectors too. A list is created which stores the vector indices present in the given cluster. This marks completion of one iteration. Dot product of each unit vector with every level means is calculated and the cluster mean which gets the maximum dot product stores the vector i.e. the list storing the indices is updated. New means of the updated clusters are calculated and are converted to unit mean vectors. Dot products of the new unit mean vectors and the previous unit mean vectors are calculated. If dot product values of all mean vectors are greater than threshold (In our code we used 0.99 as threshold), the process is terminated, else the next iteration starts and the whole process repeats.

INTRODUCTION TO PROBLEM :   
  
We will be given the Data of ‘N’ peoples each having ‘Ni’ number of images. We need to create the Dictionary of these data. So we need to create this Dictionary by using K- Means clustering, we also need to do by covariance Matrix computation and M-Spherical Means.  
  
Figure :   
  
Block Diagram Representation of the Project  
  
Proposed approach :   
Our approach is first we need to load the data of all the images. And we need to vectorize these images and we need to do K – means clustering and then we need to do covariance matrix computation, M – spherical means to create our Dictionary of size M.

Report Organization :   
  
Our report is organised as follows :   
  
1. Title and authors  
  
2. Abstract  
  
3. Introduction to problem  
  
4. Proposed approach  
  
5. Experimental Results, Project Code and Discussions  
  
6. Conclusions and summary  
  
  
  
Proposed approach :   
  
We have 80 Folders of data and we read 60 folders and stored them in a file ‘project\_input.mat(x)’  
It takes some time as it reads input from all the folders. We also vectrorised all the images into rows.   
  
example of row vector of a image.  
  
So when we load project\_input .mat, it loads all the images into matrix-‘X’, such that each row is an image signal.   
Here later we need to do K-Means clustering of these images, here we choosed k = 3, as 3 is the best predicted K by Elbow method. We can change K as we wish but we get some other dictionary.  
 Now we need to apply K-Means clustering, Here best\_result .m initialises centroids 30 times randomly. We find centroids and indexes using MYkmeans for each and every initialization and choosing the one which gives minimum distortion value( cost function ). Here [ centroid , idx , J] = best\_result (X,K) initialises centroids 30 times and does iterates 15 times for each initialisations. K-centroids and idx stores indices of each signal (1 to k). 'bestcentroids3.mat’ stores the best centroid and ‘ bestidx3.mat ’ stores the best idx.  
 separating clusters and finding eigen vectors and :  
  
  
COV\_E\_cell = cell (K,4), this cell stores Kth cluster corresponding eigen vectors, eigen values and important eigen vectors in Kth-row.  
EGN gives matrix of eigen vectors and matrix of eigen values.Here we choosed first ‘n’ eigen vectors using ‘select\_keign.m’ so that 99.3% of variance was retained and storing them in 4th column of cell corresponding to that cluster row. ‘select\_keign.m’ function finds n-eigen vectors, COV\_E\_cell contains 1st column – clusters , 2nd column – eigen vectors , 3rd column – eigen values and 4th column – first ‘n’ eign vectors correspondingly. By appending all those first ‘n’ eigen vectors from all clusters we will get an over complete dictionary, based on that ‘retaining variance’ assumed, we may not get an overcomplete so that can be a under complete or complete dictionary. So we need to choose wisely , here in our case we assumed retaining variance = 99.3%. In our case ‘D’ contains more atoms than the dimension of an image signal, so this can be called as an “ OVER COMPLETE DICTIONARY ”.

Data set Discription :   
  
Our data consists of 101 folders and each folders have 100 – 200 images. Each folder has Images of an individual person. All the images of the persons are faces having different expressions and taken in a different angles. The total number of images we used are more than 6,000. The link of the data can be found -   
  
Discussion :   
  
To get Over complete dictionary we used retaining variance 99.3%. In our case Dictionary contains more atoms than the dimension of an image signal so our dictionary is ‘ Over Complete Dictionary ’ . To find out the best K for K – means clustering we used elbow method. By using elbow method we found the best K as 3.  
  
  
Codes :   
  
input\_read.m - Reads all the data of images in the folder.  
  
imagepath.m – vectorises the images into rows or columns.  
  
best\_result.m – Initialises the centroids.  
  
Kmeans\_initialisation.m – Initialises Kmeans.  
  
  
  
bestcentroids3.mat – Stores best centroid.  
  
bestidx3.mat – Stores best index.  
  
EGN – Gives matrix of eigen vectors.  
  
select\_keigen.m – Selects n eigen vectors.  
  
dic\_project.m – This is our complete project script.  
  
  
Summary :   
  
We created the Dictionary of images of the persons by using K – Means clustering, Covariance Matrix computation and M – spherical means.  
  
Future Extension :   
  
Dictionary learning has many applications in our present world.