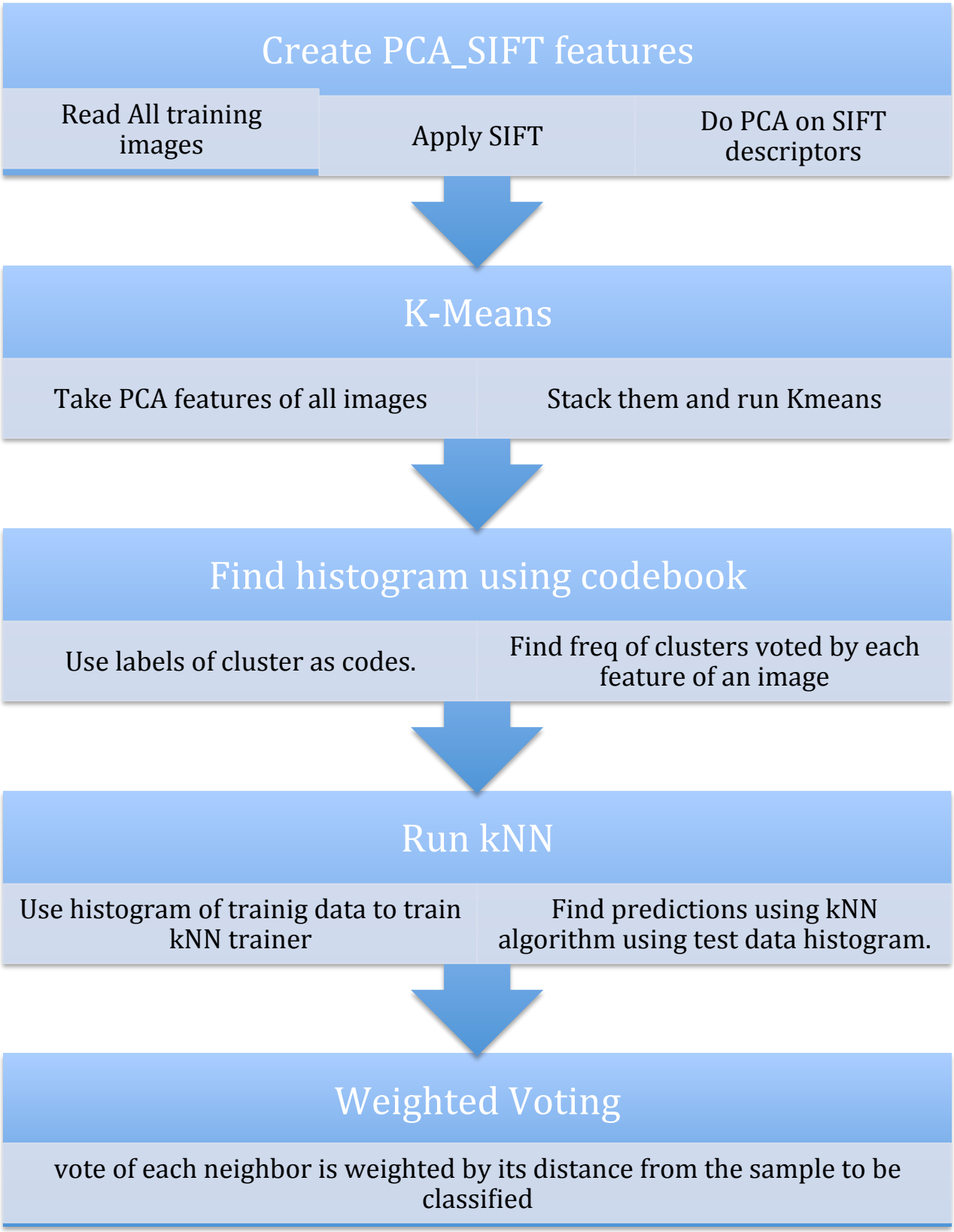
Table of Contents

Steps followed in Algorithm



All Results for one of the case

Let's take dummy case in which I have deliberately trimmed down the parameters so as for easy visualization.

Consider the following configurations:

PCA Dimension: 20
Number of k clusters: 8
Number of kNN neighbours: 10

1.) PCA_SIFT Features

First of all, the entire training set of images are read. SIFT is applied on all these images one by one. PCA is applied on top of these SIFT descriptors , while stacking PCA progressively. The idea is to combine data set for k-means algorithm.

```
[[  7.91375580e+01  -1.36237457e+02   4.99976540e+00   8.27981873e+01
    5.14168596e+00   5.29466591e+01  -2.36506157e+01   1.71296646e+02
   -4.98748245e+01   1.76415195e+01   3.56541710e+01  -8.98457718e+01
   -3.01207485e+01   6.19378281e+01  -4.57433790e-01   5.72350235e+01
    7.03148804e+01   2.00585976e+01   5.46247673e+01  -9.63293463e-02]
 [  1.23478394e+01  -2.57963440e+02  -1.22265274e+02   1.13528320e+02
    3.83266716e+01   2.60507889e+01   1.31830673e+01  -5.94795227e+01
   -1.11653305e+02   1.53928040e+02   5.21587563e+01  -5.96044617e+01
   -4.24071846e+01  -1.18413982e+01   5.20144463e+01  -4.85612535e+00
   -3.81745338e+01   1.20319168e+02  -6.18089828e+01  -3.07918282e+01]
 [ -1.77747864e+02   2.16794754e+02   1.94587219e+02   1.11248436e+02
   -5.59287567e+01   4.36763000e+01   8.43215256e+01   2.99868374e+01
   -4.99895744e+01   5.49332275e+01   5.07387695e+01   5.08298416e+01
    1.73155272e+00  -1.63026886e+01  -2.24548550e+01   1.61158924e+01
    3.67790146e+01  -9.13895321e+00  -2.42058525e+01  -3.49414301e+00]
 [ -6.95835876e+01   2.78587830e+02  -1.12778419e+02   9.02120056e+01
    9.86508331e+01   3.26791611e+01   2.82646751e+01  -6.62531738e+01
   -5.74127197e+01   4.81772537e+01  -2.72558231e+01   1.07929850e+01
   -2.57068214e+01  -6.82328033e+00  -2.40130882e+01   9.44993744e+01
    3.59929161e+01  -4.62834129e+01  -4.77200394e+01   2.68908939e+01]
 [  2.04379074e+02   9.68407593e+01  -4.51276207e+00  -1.55169811e+01
    2.92576561e+01   2.88143845e+01  -1.92979660e+01   1.04206644e-01
    2.36820068e+01  -2.07360859e+01  -3.01220932e+01  -4.56458244e+01
   -3.02148676e+00   3.04173317e+01  -8.53530788e+00   1.27653390e-01
   -7.49950266e+00   1.10096407e+00   6.48902893e+00   7.78639650e+00]]
```

The above figure shows pca_shift_features of airplanes/train/image_0071.jpeg

2.) K-means.

Next, we apply on the stack of pca_sift_features formed in the step above. Below are the centers we find. We find 8 centers each of dimension 1X 20.

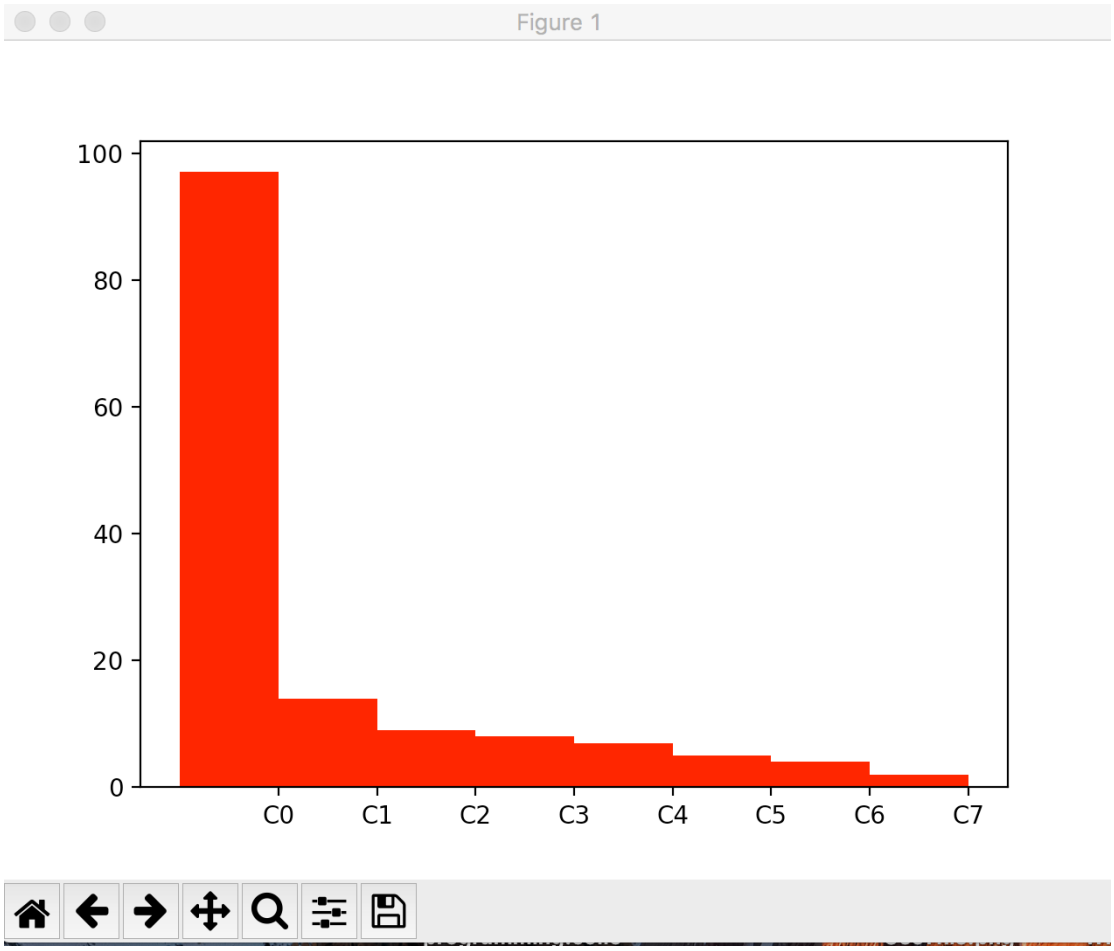
```
Centers:  [[ -1.24487930e+02  -6.57443924e+01  -1.26924372e+01   8.63167267e+01
   -3.96912880e+01  -6.02306175e+00  -3.73446345e-02  -3.96340442e+00
   -1.11546016e+00  -5.84463692e+00  -1.62297010e+00   1.75403094e+00
    5.77813745e-01  -1.69955039e+00   1.07935786e-01   3.53842318e-01
    6.84138358e-01  -1.51731551e+00   1.32521963e+00   1.37884524e-02]
 [  1.09987961e+02  -1.04580986e+02  -2.61651688e+01   4.55061493e+01
    3.30545502e+01   6.81078374e-01  -2.72515202e+00   1.07342172e+00
   -3.39087427e-01   1.83661878e+00   2.73245502e+00   1.22918952e-02
    5.62667549e-01   1.22933197e+00   2.71981627e-01   1.15568101e+00
   -2.52594829e+00   2.08503485e+00  -7.63939977e-01   1.97254014e+00]
 [  5.69391403e+01  -6.86504059e+01  -5.48284378e+01  -7.41917725e+01
   -7.27845154e+01   2.03398561e+00  -3.36742616e+00  -3.33003664e+00
    2.18349314e+00   3.98395753e+00   7.19377935e-01   3.49479318e+00
   -1.50302625e+00   9.52294528e-01   3.85259926e-01   1.22825265e+00
    8.49702954e-01   2.34592104e+00  -2.83183247e-01  -3.81741023e+00]
 [ -1.21002998e+02  -7.96321564e+01  -2.24549313e+01  -6.84736786e+01
    5.47465057e+01  -2.03692889e+00   8.16486895e-01   3.56024694e+00
    1.48978341e+00  -9.32777286e-01  -3.29560781e+00  -3.90789008e+00
   -1.58481851e-01  -1.57596326e+00  -1.20277941e+00  -3.27271628e+00
    1.70175767e+00  -2.09381771e+00  -5.66136800e-02   1.42828906e+00]
 [  9.34501724e+01  -1.56041603e+01   1.38153336e+02  -1.46567497e+01
    4.42642879e+00   3.87316918e+00   6.00189543e+00   1.17414856e+00
   -1.02096832e+00  -5.41123509e-01   4.32490781e-02  -1.62995207e+00
    1.38350213e+00  -2.55925608e+00   1.00423920e+00  -6.16448939e-01
    1.69945204e+00  -2.76600862e+00   4.47285563e-01  -6.65180147e-01]
 [  1.45605713e+02   1.07327599e+02  -2.23225918e+01   5.16782522e+00
    7.73995101e-01  -1.82845592e+00  -2.08611444e-01   8.86536948e-03
   -7.74140596e-01  -3.18164897e+00  -1.48738110e+00   3.12631458e-01
   -7.46373951e-01   6.71697199e-01  -9.65424716e-01  -1.30939627e+00
    3.31621885e-01  -1.49564207e+00  -1.25550143e-02   3.94259065e-01]
 [ -6.08883286e+01   1.27578163e+02  -7.97200928e+01   6.53908634e+00
    9.71728897e+00   1.40996540e+00   1.13768673e+00   2.18085051e-01
    1.73712838e+00   4.48949242e+00   2.33389091e+00  -1.07823002e+00
    3.07369739e-01  -1.13921988e+00   6.36284292e-01   1.56196618e+00
   -1.36927426e+00   9.12417769e-01   6.91554189e-01  -4.85153109e-01]
 [ -1.37533401e+02   8.01870880e+01   9.56420212e+01  -7.65257883e+00
   -5.25419998e+00   2.79428291e+00  -1.24283433e+00   7.72563457e-01
   -1.77278686e+00   1.05481970e+00   4.99683559e-01   1.36464250e+00
   -4.90021914e-01   3.85961199e+00   2.74964496e-02   1.08588147e+00
   -7.73588002e-01   2.69867826e+00  -1.29239106e+00   2.05138087e-01]]
```

3.) Create histograms using codebook.

In this step, basically we create histogram using image features and codebook. Codebook is the assigned codes to k clusters by algorithm. Intention is to find what is frequency of cluster supported by respective descriptor of an image.

Here is the sample of histogram example of one of the training image. That is: airplanes/train/image_0071.jpeg

Histogram achieved for this image was [97, 14, 9, 8, 7, 5, 4, 2]. Such that value at each ith index is the frequency of cluster ‘i’ supported by feature of the above mentioned training image. It can be visualized as below image



4.) KNN and predicting

Last step is to do kNN analysis on histograms achieved in the step above and next is to predict labels.

Prediction achieved for 50 test images is given as:

[0, 0, 1, 4, 1, 0, 1, 3, 3, 3, 1, 2, 2, 1, 2, 0, 1, 2, 3, 0, 0, 1, 3, 3, 0, 0, 4, 1, 1, 2, 2, 3, 1, 3, 1, 0, 1, 0, 0, 3, 0, 2, 0, 0, 3, 3, 1, 3, 1, 4]

Screenshot:

```
Prediction labels: [3, 1, 2, 1, 1, 1, 0, 0, 1, 0, 1, 3, 1, 1, 1, 1, 0, 1, 0, 2, 0, 0, 3, 1, 1, 0, 3, 0, 0, 3, 3, 2, 4, 4, 1, 0, 2, 1, 0, 4, 3,
3, 2, 1, 0, 3, 1, 1, 1, 0]
Correctly classified:  20.0 %
```

5.) Final step – Accuracy

The last step of the algorithm is to compute accuracy. In this case, since clusters were very small hence accuracy is 20.0% (Screenshot above.)

6.) Proof of correctness

When I ran the algorithm on various combinations of key parameters that is PCA dimension, K clusters, number of neighbors in kNN, I found that accuracy was 100% in most cases. Even with such a few number of clusters, achieved accuracy by using training dataset as test dataset is 99%.

Screenshot:

```
(opencv) Deepikas-MacBook-Pro-2:Assign5 deepika$ python -W ignore Assign5.py
Parameters.....
PCA Dimension:  20
Number of k clusters:  8
Number of kNN neighbours:  10
Reading training images......
Clustering in progres.....
Computing test images codewords and histograms.....Done!
Starting Categorization via kNN Algorithm...Done!
Predicting....
Prediction labels: [0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2,
2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 4, 4, 4, 4, 4, 4, 4, 4,
 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4]
Correctly classified:  99.0 %
(opencv) Deepikas-MacBook-Pro-2:Assign5 deepika$
```

Results with various combinations

```
Parameters.....
PCA Dimension:  20
Number of k clusters:  100
Number of kNN neighbours:  10
Reading training images......
Clustering in progress.....
Computing test images codewords and histograms.....Done!
Starting Categorization via kNN Algorithm...Done!
Predicting....
Correctly classified:  20.0 %
```

```
Parameters.....
PCA Dimension:  20
Number of k clusters:  125
Number of kNN neighbours:  10
Reading training images......
Clustering in progress.....
Computing test images codewords and histograms.....Done!
Starting Categorization via kNN Algorithm...Done!
Predicting....
Correctly classified:  20.0 %
```

```
Parameters.....
PCA Dimension:  20
Number of k clusters:  150
Number of kNN neighbours:  10
Reading training images......
Clustering in progress.....
Computing test images codewords and histograms.....Done!
Starting Categorization via kNN Algorithm...Done!
Predicting....
Correctly classified:  26.0 %
```

```
Parameters.....
PCA Dimension:  20
Number of k clusters:  100
Number of kNN neighbours:  15
Reading training images......
Clustering in progress.....
Computing test images codewords and histograms.....Done!
Starting Categorization via kNN Algorithm...Done!
Predicting....
Correctly classified:  24.0 %
```

```
Parameters.....
PCA Dimension:  20
Number of k clusters:  100
Number of kNN neighbours:  20
Reading training images......
Clustering in progress.....
Computing test images codewords and histograms.....Done!
Starting Categorization via kNN Algorithm...Done!
Predicting....
Correctly classified:  22.0 %
```

```
Parameters.....
PCA Dimension:  20
Number of k clusters:  100
Number of kNN neighbours:  40
Reading training images......
Clustering in progress.....
Computing test images codewords and histograms.....Done!
Starting Categorization via kNN Algorithm...Done!
Predicting....
Correctly classified:  18.0 %
```

```
Parameters.....
PCA Dimension:  50
Number of k clusters:  100
Number of kNN neighbours:  10
Reading training images......
Clustering in progress.....
Computing test images codewords and histograms.....Done!
Starting Categorization via kNN Algorithm...Done!
Predicting....
Correctly classified:  28.0 %
```

```
Parameters.....
PCA Dimension:  50
Number of k clusters:  120
Number of kNN neighbours:  10
Reading training images......
Clustering in progress.....
Computing test images codewords and histograms.....Done!
Starting Categorization via kNN Algorithm...Done!
Predicting....
Correctly classified:  20.0 %
```

```
Parameters.....
PCA Dimension:  55
Number of k clusters:  120
Number of kNN neighbours:  12
Reading training images......
Clustering in progress.....
Computing test images codewords and histograms.....Done!
Starting Categorization via kNN Algorithm...Done!
Predicting....
Correctly classified:  20.0 %
```

```
Parameters.....
PCA Dimension:  40
Number of k clusters:  100
Number of kNN neighbours:  10
Reading training images......
Clustering in progress.....
Computing test images codewords and histograms.....Done!
Starting Categorization via kNN Algorithm...Done!
Predicting....
Correctly classified:  22.0 %
```

```
Parameters.....
PCA Dimension:  40
Number of k clusters:  100
Number of kNN neighbours:  20
Reading training images......
Clustering in progress.....
Computing test images codewords and histograms.....Done!
Starting Categorization via kNN Algorithm...Done!
Predicting....
Correctly classified:  30.0 %
```

```
Parameters.....
PCA Dimension:  40
Number of k clusters:  120
Number of kNN neighbours:  15
Reading training images......
Clustering in progress.....
Computing test images codewords and histograms.....Done!
Starting Categorization via kNN Algorithm...Done!
Predicting....
Correctly classified:  14.0 %
```

```
Parameters.....
PCA Dimension:  40
Number of k clusters:  100
Number of kNN neighbours:  25
Reading training images......
Clustering in progress.....
Computing test images codewords and histograms.....Done!
Starting Categorization via kNN Algorithm...Done!
Predicting....
Correctly classified:  24.0 %
```

```
Parameters.....
PCA Dimension:  40
Number of k clusters:  100
Number of kNN neighbours:  25
Reading training images......
Clustering in progress.....
Computing test images codewords and histograms.....Done!
Starting Categorization via kNN Algorithm...Done!
Predicting....
Correctly classified:  100.0 %
```

Results in summary (tabular form)

PCA Components	K clusters	Neighbors in kNN	Accuracy
20	100	10	20%
20	125	10	20%
20	150	10	26%
20	100	15	24%
20	100	20	22%
20	100	40	18%
50	100	10	28%
50	120	10	20%
55	120	12	20%
40	100	10	22%
40	100	20	30%
40	120	15	14%
40	100	25	24%
40	100	25	100% (Using training data for testing)

Comments

The algorithm accuracy varies depending on centers chosen by k-means. the goodness of center decides the accuracy of the algorithm . The best accuracy I could achieve was using PCA components = 40, clusters = 100 and k-nearest neighbors as 20. Each of these parameters have peek. That is increasing each of the key parameters till a point increased accuracy and then suddenly drops down. So, a lot of tuning and trial/error analysis is required to come up with high accuracy. This is one of disadvantage of this algorithm as parameters need to be tuned depending on input data set.

K-means cannot handle non-linear functions. So, if any of the images had a non-linear function that would have been less favored in predictions.