COMPUTER VISION ASSIGNMENT 5

K value	Model	Accuracy(%)
300	Colour Histogram and KNN	43
300	Bag of Words and KNN	50.88
300	Bag of Words and SVM	57.87
300	Bag of Words ,SVM, Soft Assignment	63.63
300	Bag of Words and Spatial Pyramid, SVM	65.37
300	GMM Clustering and fisher encoding	68

1.COLOUR HISTOGRAM AND KNN CLASSIFIER

Method:

1. The bin size is computed by dividing the total pixels by the number of bins. The number of bins chosen was 50.
2. The three channels were extracted from the image. It was divided by the bin size and the joint colour matrix was formed by summing them.

```
im = double(img(:,:,i));
im = reshape(im1,[h*w 1]);
binnumber = floor(im1/bin_size(i));
```

3. The joint pixel matrix was then divided into a histogram by hist function. This histogram was normalized.

```
hist_bin = hist(temp,1:c) + 1;
h = hist_bin/sum(hist_bin);
```

4. The histogram was constructed for every training image. Similar histogram was constructed for every test image. The distance between every histogram of test image and the training image was calculated using pdist2. The value of k chosen was 50. The label corresponding to the training image which was closest was assigned to the test image.

```
for k=1:1888
        temp=pdist2(hist{k},  hist_test);
        dist(k)=norm(temp);
 end
    [min_d,ind]=min(dist);
```

The accuracy achieved was 43%.

```
acc =

    0.4314
```

The below image gives the Confusion Matrix

```
confusion_matrix =

    14    47     2    13     2     6     3    13
     2    81     1     3     2     5     2     4
     3    22    49     5     7     4     4     6
     3    29     1    43     4     5    10     5
     5    46     2    10    21     5     4     7
     4    60     2     4     2    17     5     6
     1    14     1    18     5     1    53     7
     3    36     1    16     7     8     5    24
```

2.BAG OF WORDS MODEL AND KNN CLASSIFIER

Method:

1. The descriptors are initialized from the given set of training descriptors. 10 features were chosen randomly for each training image and concatenated to form 128*18880 matrix.
```
for i=1:1888
    des=train_D{i};
    t=randi(size(des,2),[1,num_per_img]) ;
    dict=horzcat(dict,des(:,t));
end
```

2. The K-means clustering is performed on the 128*18880 matrix for k=300.
   i) Random 300 features were chosen as centres
   ```
   t=randi(size(data,1),[1,k]);
   ```

   ii) Each data point is assigned to the closest cluster centre based on Euclidean distance.
   ```
   for i=1:size(data,1)
           [D,I] = pdist2(clust,data(i,:),'euclidean','Smallest',1);
           new=horzcat(new,I);
   end
   ```

   iii) The data points belonging to a particular label were averaged to find the new cluster centres.
   ```
   for j=1:k

           ind=find(new==k);
           c=mean(data(ind,:));
           clust(k,:)=c;
   end
   ```

iv) The above steps are repeated till the distance to the cluster converges and hence an encoded dictionary of size 128*300 is formed.

3. This dictionary is used for creating the train_data and test_data. Each descriptor is compared with these cluster centres. Each centre in the dictionary is like a word consisting 300 words in this case. Based on the closest distance each descriptor is assigned a word. Now a histogram is formed by voting the number of SIFT features belonging to each word. This histogram is normalized.

```
for u=1:num_train
    temp=double(train_D{u}');
    ind=[];
    for j=1:size(temp,1)
         [D,I] = pdist2(clust,temp(j,:),'euclidean','Smallest',1);
         ind=horzcat(ind,I);
    end
    size(ind);
    ind1=knnsearch(clust,temp,'K',300);
    size(ind1)
    train_data_new(u,:)=histc(ind,1:k);
    train_data_new(u,:)=train_data_new(u,:)/sum(train_data_new(u,:));
end
```

4. A similar procedure is followed for creating the test_data matrix.
5. Classification: The KNN classifier is used. The index corresponding to the data in test_data that is closest to train_data is evaluated. The train_gs label of this index is assigned as the label to the test image.

```
for u=1:800
    [m,ind]=min(sum(abs(train_data-repmat(test_data(u,:),[num_train,1])),2));
    labels(u) = train_gs(ind);
end
```

K-Means stopping criterion:
The K-Means clustering is performed till the distance between the old cluster centre and the current cluster centre converges less than 0.1

```
while(abs(old_clust-clust))>0.01)
```

The efficiency for Bag of Words model with KNN and K=300 is 50.88%.

```
confusion_matrix =

    46     0    21     3     9    19     2     0
     1    89     0     0     5     3     1     1
    18     1    53     1    16     7     2     2
     0     9     3    46     7     6    13    16
     8    14     3     0    56     9     6     4
    21    10     4     1    13    39    11     1
     3    11     5     7    18     9    42     5
    12     3     5    20    12     6     6    36


accuracy =

    0.5088
```

3.BAG OF WORDS MODEL AND SVM CLASSIFIER

Method:

1. The bag of words model, encoding and the feature representation of training and testing data is same as done in the previous model.
2. Classification: The SVM classifier was used using fitcecoc. The train_data and the test_data created were used to create the model and predict the labels.

```
t = templateSVM('Standardize', 1);
model=fitcecoc((train_data),(train_gs'),'Learners', t);
labels=predict(model,test_data);
```

The accuracy achieved is 57.87% for k=300.

```
accuracy =

   57.8750


t1 =

    1.3639


t2 =

    0.2677
```

The t1 gives the training time and t2 gives the testing time using svm classifier in seconds.

```
confusion_matrix =

    62     0    11     0    11    14     1     1
     0    79     0     0     7     5     7     2
    22     0    45     1    12     8     7     5
     1     2     2    51     3     2    14    25
     8     3     4     1    58    17     6     3
    19     3     7     0    16    50     2     3
     2     2     2     9     6     7    56    16
     0     0     4    18     5     1    10    62
```

Confusion Matrix

4.SPATIAL PYRAMID MODEL AND SVM CLASSIFIER

Method:

1. The image is sub divided according to the number of levels. The number of levels chosen was 1. Hence the image was sub divided into 4 sub regions.
2. The train_data set is constructed as same way done in the previous problems. However here the descriptors for individual sub regions are chosen according to the coordinates given in the train_F matrix. The corresponding descriptors lying in that region are used to construct the train data matrix. Thus four 300*1888 matrices are formed each belonging to a sub region. These are concatenated with the train_data matrix for the actual image. Thus the train_data matrix will be 1500*1888 size.
3. The test_data is calculated the same way for all test images. This will lead to a matrix of size 1500*800.

```
for n=1:num_test
          nm=sprintf('C:/First_sem/CV/hw5/test/%d.jpg',n)
    im=(imread(nm));
    ans=size(im);
    [r,c,z]=size(im);
    ha=r;
    wa=c;
    a=floor(ha/2);
    b=floor(wa/2);
    temp=test_F{n};
    if sub==1
            x_ind=find(temp(1,:)<b);
            y_ind=find(temp(2,:)<a);
    elseif sub==2
          x_ind=find(temp(1,:)>b&temp(1,:)<wa);
          y_ind=find(temp(2,:)<a);
    elseif sub==3
            x_ind=find(temp(1,:)<b);
            y_ind=find(temp(2,:)<ha&temp(2,:)>a);
    else
          x_ind=find(temp(1,:)>b&temp(1,:)<wa);
          y_ind=find(temp(2,:)<ha&temp(2,:)>a);
        end

    if n==193
        index=y_ind(3:end);
    else
        index=intersect(x_ind,y_ind);
    end
    test_D_new{n}=test_D{n}(:,index);
end
 size(test_D_new)
 save test_D_new
for u=1:800
    temp=double(test_D_new{u}');
    ind=[];
    for j=1:size(temp,1)
         [D,I] =
pdist2(clust,temp(j,:),'euclidean','Smallest',1);
         ind=horzcat(ind,I);
    end
test_data_new_sub_temp(u,:)=histc(ind,1:k);

test_data_new_sub_temp(u,:)=test_data_new_sub_temp(u,:)/sum(test
_data_new_sub_temp(u,:));
end
```

An accuracy of 65.37% was achieved for k=300.

```
accuracy =

   65.3750
```

```
t1 =

    4.2200


t2 =

    0.4443
```

T1 and t2 gives the training and testing time correspondingly.

EXTRAS:

1. SOFT ASSIGNMENT

1. The dictionary is formed using the k means clustering. However, here the encoding procedure is different from histogram. Once when the nearest neighbours are found for every feature, instead of assigning one feature to a word(which is hard assignment), here a single feature will be allocated to every word with a probability determined by the gausian kernel.

$$K_\sigma(x) = \frac{1}{\sqrt{2\pi\sigma}} \exp\left(-\frac{1}{2}\frac{x^2}{\sigma^2}\right) .$$

Hence this is soft assignment.

```
d_new=sort(dis,'descend');
variance= mean(d_new(1:5));
w=exp(-0.5*(dis.^2)./variance);
weight=sum(w,2);
train_data_soft(u,:)=weight';
train_data_soft(u,:)=train_data_soft(u,:)/sum(train_data_soft(
u,:));
```

This achieved greater efficiency than hard assignment and gave 63%. for k=300 compared to 57% which was achieved with hard assignment. T1 and t2 gave the training and testing times for the svm classifier.

```
accuracy =

          63.63


t1 =

           2.80


t2 =

           0.10
```

2. GMM DISTRIBUTION AND FISHER ENCODING

Method:

1. The clustering is performed using GMM distribution.

```
obj=fitgmdist(data,300,'CovarianceType','diagonal','Regularizati
onValue',0.1)

clust=obj.mu
```

2.The soft assignment encoding is done to form the test and training data according to the equation:

```
w=exp(-0.5*(dis.^2)./variance);
weight=sum(w,2);
train_data_soft(u,:)=weight';

train_data_soft(u,:)=train_data_soft(u,:)/sum(train_data_soft(
u,:));
```

```
confusion_matrix =

    64     0     5     0     9    19     2     1
     0    84     0     0     6     6     4     0
    19     0    54     4     8     8     5     2
     0     0     1    71     1     3    11    13
     6     3     0     0    67    16     4     4
    11     2     5     0     9    71     1     1
     0     1     3    10     7     3    65    11
     1     2     2    10     3     3    11    68


accuracy =

    68
```

3. ACCURACY FOR DIFFERENT K VALUES AND BAG OF WORDS MODEL

As it can be seen, when the bag of words model with SVM classifier was tried for different K values, the accuracy increased as the value of k increased.

```
confusion_matrix =

    61     1     9     1    11    12     0     5
     0    77     0     2    11     0    10     0
    43     0    14     2     6    21     3    11
     0     3     0    47     2     9    14    25
     1     4     1     2    51    20    10    11
    13     2     2     3    15    48     7    10
     1     6     0    18    16    11    35    13
     5     6     1    13    14    17    12    32


accuracy =

   45.6250
```

Accuracy for k=25

```
confusion_matrix =

    52     0    12     2    13    16     1     4
     0    74     0     3    12     3     4     4
    33     1    30     6     6    15     1     8
     0     2     1    60     5     3    11    18
     2     6     3     1    43    22    11    12
    17     0     1     3    17    52     6     4
     1     4     2     8     5    16    48    16
     3     3     3    11    15     7    13    45


accuracy =

   50.5000
```

Accuracy for k=50

```
confusion_matrix =

    56     0    10     0    10    20     2     2
     0    70     0     3     7     6    10     4
    22     0    33     1    14    21     4     5
     0     3     0    49     3    11     9    25
     5     2     1     1    47    26     6    12
    17     0     2     4    15    53     6     3
     3     2     1     9    15    13    33    24
     2     2     4    13     5     7     8    59


accuracy =

    50
```

Accuracy for k=100

```
confusion_matrix =

    52     0    11     0    11    22     3     1
     0    71     1     0    14    11     2     1
    25     0    43     2     5    14     6     5
     0     1     0    57     3     7    11    21
     7     2     4     0    50    21    11     5
    13     1     6     1    10    59     3     7
     0     1     0     9    16    10    44    20
     3     4     3    18     8     4     6    54


accuracy =

   53.7500
```

Accuracy for k=200

```
confusion_matrix =

    56     0    10     0    11    20     1     2
     0    77     0     2     9     8     2     2
    21     0    43     0     7    17     8     4
     0     2     0    57     2     3    11    25
     4     2     2     0    46    35     8     3
    16     1     7     1    18    53     2     2
     0     0     2     7    10     7    56    18
     1     1     2    12     5     6    14    59


accuracy =

   55.8750
```

Accuracy for k=400

```
confusion_matrix =

    58     0     8     0     8    23     2     1
     0    79     0     0     8     6     4     3
    23     0    36     2     7    18    10     4
     0     1     0    60     2     5    12    20
     6     2     2     0    62    21     5     2
    13     2     4     0    24    51     3     3
     1     0     1     6    12     8    62    10
     2     0     4    10     2     4     8    70


accuracy =

   59.7500
```

Accuracy for k=800

```
confusion_matrix =

    56     0     7     0    10    23     2     2
     0    75     0     0    15     3     7     0
    21     0    41     2    10    15     7     4
     0     0     0    54     1     4    16    25
     5     1     1     0    65    20     6     2
    11     0     1     1    18    65     3     1
     0     0     0     6    14     7    64     9
     1     1     4    10     6     3    12    63


accuracy =

   60.3750
```

Accuracy for k=1600