CVIP HW2 Report (by Gaurav Kshirsagar)

Overview:

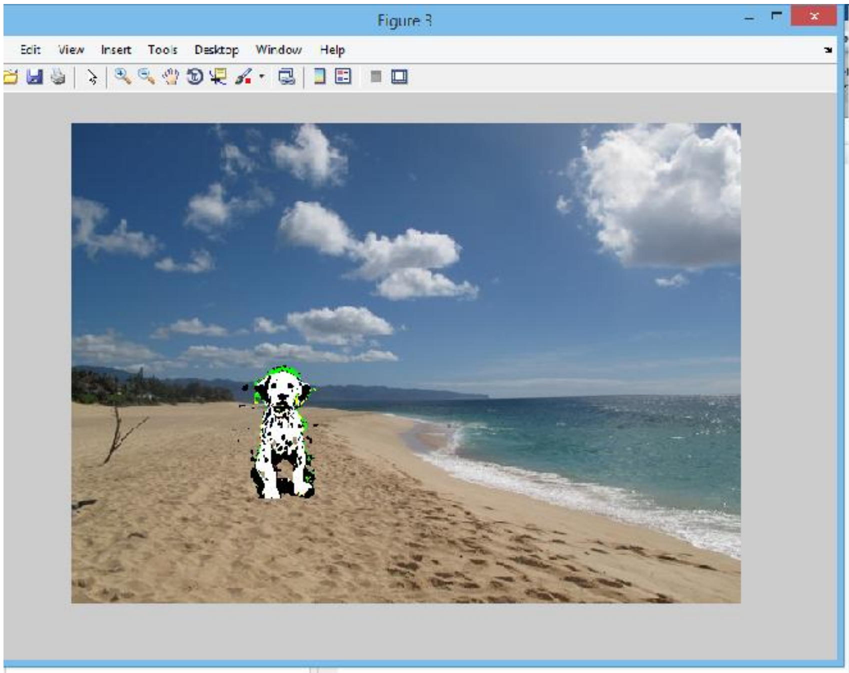
The image of the animal (eg. ‘dog.jpg’) is taken and converted to double precision and to grayscale image. This is ‘grayI’ in the code. **All this is done in a run file I have included ‘run_me.m’.** In the segmentation part, this image is convolved with the filter array and we get ‘filter_responses’ which is the result of applying the 48 filters. Then this is converted to all positive values.

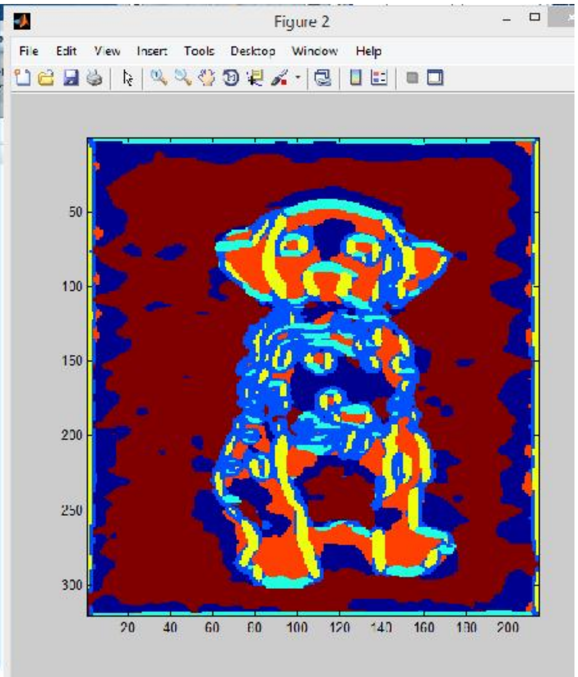
Now to apply K-means on it, we use ‘X’ which is the array [height*width,48] after reshaping the above result. Then the result ’IDX’ of K-means is reshaped into an array [height,width,48] again and displayed with the K segments generated by it. **I have used the K-means provided and not made my own code for it.** (I have tried some K means implementation but its not complete or tested).

From ‘IDX’ we need to get only the clusters representing the foreground which are defined by the ‘fgs’ array. No. of elements in it can be at most K. I have manually specified the values like fgs=[1 2 3 5] if some K is at least 5.

Then the original image is read in ‘actualI’ and new background image is in ‘newbg’. Both are converted to double and transferImg(fgs,IDX,actualI,newbg) is called to move the foreground part to the new background. Finally, the resultant image ‘newI’ is displayed.

Sample outputs for dog and cheetah:





K=6

fgs= [1 2 3 4 5]

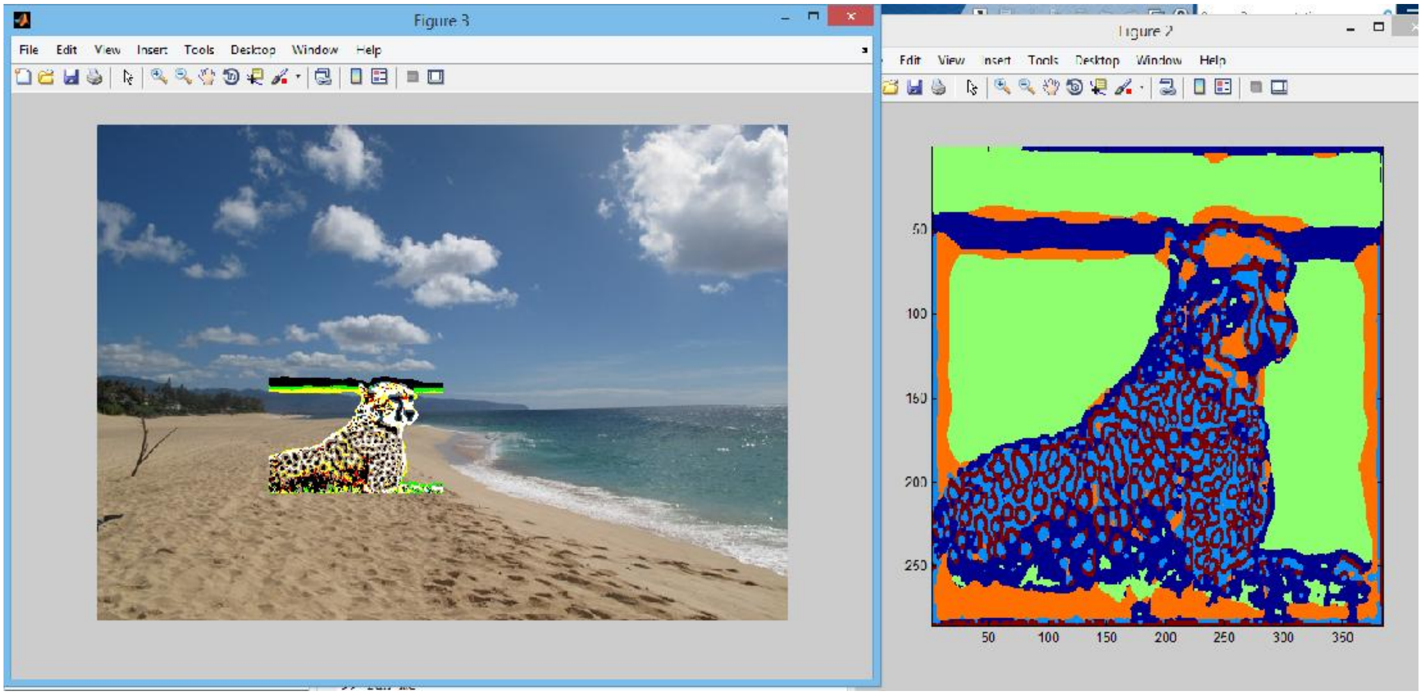


Figure 3
File Edit View Insert Tools Desktop Window Help
Figure 2
Edit View Insert Tools Desktop Window Help
50
100
150
200
250
50
100
150
200
250
300
350

K=5

fgs= [1 2 4]

Conclusions and comments:

The output of K-means is different in many runs of the same code with same values which must be because of the certain randomness of K means clustering. The output above is the most consistent one I got.

There are quite a few noticeable holes in the segments which thus, appear in the final transferred image. Now, we are applying filters that highlight the texture and patterns in the original image. They are taken as important in the K-means and the image is segmented. However, some parts from the foreground also may have the same texture as some other points in background and so they too show up in the same segments. As a result of this, we are able to take a segment in the final image in which most of the points are in the foreground area. Some points clustered with the background ones cannot be taken and they show up as holes.

Improvements possible:

We can use more clusters i.e. increase value of K so that we have enough number of clusters from which we select only a few representing the part we are interested in. For the given images, I ran the code using K=10 or 15 but this prevents the K- means from converging in limited time. Also, if the size of images is larger then we can extract more features after filtration and K-means could cluster the image better.

Changes to be made to run code on different images (The code is commented as shown below to show where to edit):

- File run_me.m
    1. Give the input image here to segment @ line 4
       ```
       %image to pick animal from
       imname='images/cheetah.jpg';
       ```
    2. Give the K value @line 9
       ```
       %no. of segments
       k=5;
       ```

- File segmentImg.m
    1. Fgs array @ line 75
       ```
       %init the fgs array to select the clusters you want to show
       fgs=[1 2 3 4];
       ```
    2. Original image @ line 77
       ```
       %original image
       actualI=imread('images/cheetah.jpg');
       ```
    3. New bckg image @line 81
       ```
       %new background
       newbg=imread('images/bg.jpg');
       ```