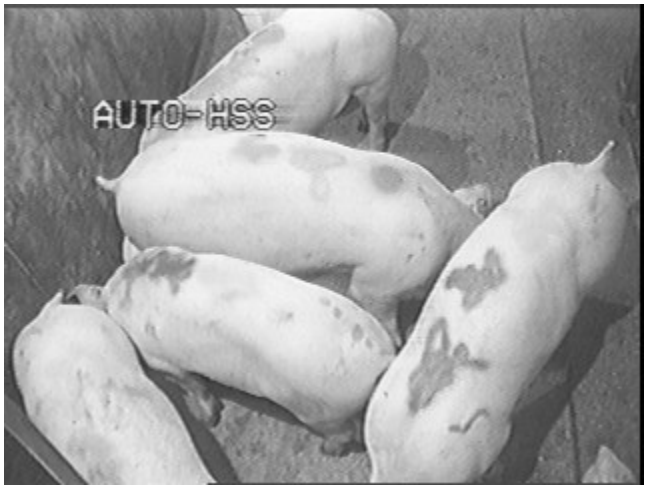
CS555/455 Fall 2018
Shriprasad Bhamare
B00713668

Assignment 5 Writeup

Programming Part A: Pattern Recognition (60%)

(1) Obtain image I (same as the one used in the assignment 4)



(2) Divide I into 4x4 disjoint subimages. Consider each subimage as a 16 dimensional pattern vector. Use the pattern vectors at the upper half of the image for training and use the lower half of the image for testing. Among the training patterns, manually group the training patterns into three clusters (classes) and attach a class label (e.g., certain gray level 0, 128, 255) to each pattern, produce image M1. For example, you may group the training patterns into three types of classes based on the average gray level G of 4x4 block:

a. Class 1: 0 <= G < 125
b. Class 2: 125 <= G < 175
c. Class 3: 175 <= G <= 255

(3) Perform NN classification of the testing patterns.

- Testing patterns are classified by NN classification. Replace each testing vector with the corresponding class label, produce image N1.

a. Replace each testing vector with the training vector, produce image N2.

b. Replace each testing vector with the average value of the corresponding training vector, and produce image N3.

c. Replace each testing vector with the average of the corresponding class, and produce image N4.

(4) Perform the manually grouping (like (2)) onto the testing patterns, produce image T1, compare N1 and T1, and compute the error rate E of the testing patterns.

Error Rate : 64.25

```
shri@shri-Inspiron-3537: ~/Downloads/assignment5/5a
shri@shri-Inspiron-3537:~/Downloads/assignment5$ cd 5a
shri@shri-Inspiron-3537:~/Downloads/assignment5/5a$ make
g++ -o 5a 5a.cpp `pkg-config --cflags --libs opencv`
shri@shri-Inspiron-3537:~/Downloads/assignment5/5a$ ./5a TestImage-even-width.bm
p
Error Rate:64.25
```

(5) Perform the k-means clustering algorithm onto the upper-half of the image I, obtain image K1, which shows three classes with three labels (e.g., 0, 128, 255).

(6) Your write-up should include a brief description of the algorithm implementation, and the resulting images M1, N1-N4, T1, K1, and error rate E.

For M1 the algorithm is consider is first divided the image into 4x4 blocks of upper half and then the average of each block is considered and according to the block average the Class label is applied (i.e. Suppose the average is 130 then it is as 125) and in this way it is applied on each block of 4x4 for the upper part of image which is training part.
And this is the result for M1.
consider the above algorithm but for the lower part of image but the class label applied is different here first we considered the average by using the mean formula which is square of distance between 2 pixels and then divide by total number of pixels in a block taking the square root of whole and this is applied for each every pixel in the upper block and then the minimum class label is replaced with it.
For N2 similar conditions which are used in N1 are used but instead of taking the class label the training vector is considered and each block is replaced with the training vector for the testing vector.
For N3 instead replacing with each block, the average of block is replaced in the testing vector from the training vector.
For N4 the block by block average with class label is considered instead of considering the block by block vector which is considered in N3.
consider the T1 part which is similar to M1 instead applying on the upper part it is applied on the bottom part of the image.
The error rate obtained for the given question considering the N1 and T1 is **64.25.**
Kmeans:

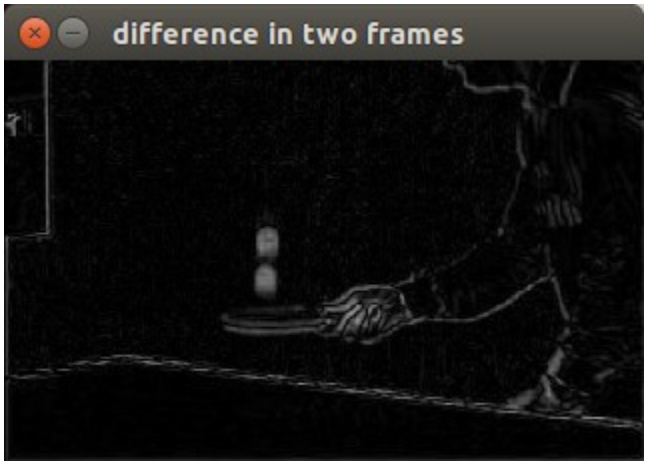
- Arbitrarily choose two samples as the initial clusters centers
- Distribute the pattern samples into the cluster domain according to the distance from the sample to the cluster center.
- Update the cluster centers
- Repeat (2) and (3) until the updated cluster centers are unchanged

-

Programming Part B: (40%) Motion Compensation and Tracking

- **Requirement (40%):** Design an algorithm to estimate motion vectors of a video sequence (e.g., using 8*8 block). You are required to (1) show your algorithm, the resulting motion vectors across image frames; (2) Show the difference of two frames; (3) Show the compensated difference of the two frames.

```
Mat src1 = imread(argv[1], 0);

Mat src2 = imread(argv[2], 0);

Mat src3;

if (src1.empty() || src2.empty())

{

        cout << "Read error" << endl;

        return -1;

 }


absdiff(im1, im2, im3);
```



In part B the motion vector is determined by using the absolute difference of two image frames 8x8 block is applied after that each block is compared with the block in the other image block if the difference is found in both of the image block the it is considered as motion vector. And on the different part the thresholding is applied so that the difference between the two images frames can easily determined.