RESEARCH ARTICLE OPEN ACCESS

Adaptive K-Means Clustering For Improving Non- Local Means Filtering

Nandini Prasad K S[1], Prabha R[2], Pushpalatha S[3]
Department of Information Science and Engg. Dr. Ambedkar Institute of Technology Bangalore, INDIA.
Department of Information Science and Engg. Dr. Ambedkar Institute of Technology Bangalore, INDIA.

Abstract
Recording devices whether analog or digital, have traits which make them susceptible to noise. In selecting a noise reduction algorithm, one must weigh several factors. Image denoising is defined as a method to recover a true image from an observed noisy image and is applied in display systems to improve the quality of image. One of the popular denoising methods, NLM, produces the quality of image compared than other denoising methods. We propose to improve non local means and using rotationally invariant block matching (RIBM) into the NLM framework. NLM applies moment invariants based K-means clustering on the Gaussian blurred image, which provides better classification before weighted averaging.
***Index Terms*—**Image, Denoising, NLM, RIBM, Clustering.

I. INTRODUCTION

An image is an array, or a matrix, of square pixels (picture elements) arranged in columns and rows. Pictures are the most common and convenient means of conveying or transmitting information. A picture is worth a thousand words. Pictures concisely convey information about positions, sizes and inter relationships between objects. They portray spatial information that we can recognize as objects. Image processing is a method to convert an image into digital form and perform some operations on it, in order to get an enhanced image or to extract some useful information from it. Image processing operations can be roughly divided into three major categories: Image Compression, Image Enhancement and Restoration, and Measurement Extraction. Image compression is familiar to most people. It involves reducing the amount of memory needed to store a digital image. Image processing involves changing the nature of an image in order to either to:
Improve its pictorial information for human interpretation, or to render it more suitable for autonomous machine perception.

Digital images types we will consider are: Binary, Gray-scale, Color and Multispectral. Binary images are the simplest type of images and can take on two values, typically black and white, or 0 and 1. A binary image is referred to as a 1-bit image because it takes only 1 binary digit to represent each pixel. Gray-scale images are referred to as monochrome (one-color) images. They contain gray level information, no color information. The number of bits used for each pixel determines the number of different gray levels available. The typical gray-scale image contains 8bits/pixel data, which allows us to have 256 different gray levels. Color images can be modeled as three-band monochrome image data, where each band of data corresponds to a different color. The actual information stored in the digital image data is the gray-level information in each spectral band. Typical color images are represented as red, green, and blue (RGB images). Multispectral images typically contain information outside the normal human perceptual range. This may include infrared, ultraviolet, X-ray, acoustic, or radar data. These are not images in the usual sense because the information represented is not directly visible by the human system. However, the information is often represented in visual form by mapping the different spectral bands to RGB components.

Types of digital image data are divided into two primary categories: bitmap and vector.
Denoising (or restoration) is still a widely studied and an unsolved problem in image processing.
Image denoising is to decompose f into two functions u, and n with $f = u + n$, where u contains the most meaningful signals depicted by f, and n represents the noise. In the ideal case, the noise part n has no any signal information.

All denoising methods depend on a filtering parameter h. For most methods, the parameter h depends on an estimation of the noise variance $\sigma 2$.
The local smoothing methods and the frequency domain filter aims at noise reduction and at a reconstruction of the main geometrical configurations but not at the preservation of the fine structure, details, and texture. Due to the regularity assumptions on the original image of previous methods, details and fine structures are smoothed out because they behave in all functional aspects as noise. Buades, Coll and Morel proposed the Non-Local (NL) means filter for image denoising. This method replaces a noisy pixel by the weighted average of other image pixels with weights reflecting the similarity between local neighborhoods

of the pixel being processed and the other pixels. The NL-means filter was proposed as an intuitive neighborhood filter but theoretical connections to diffusion and non-parametric estimation approaches.

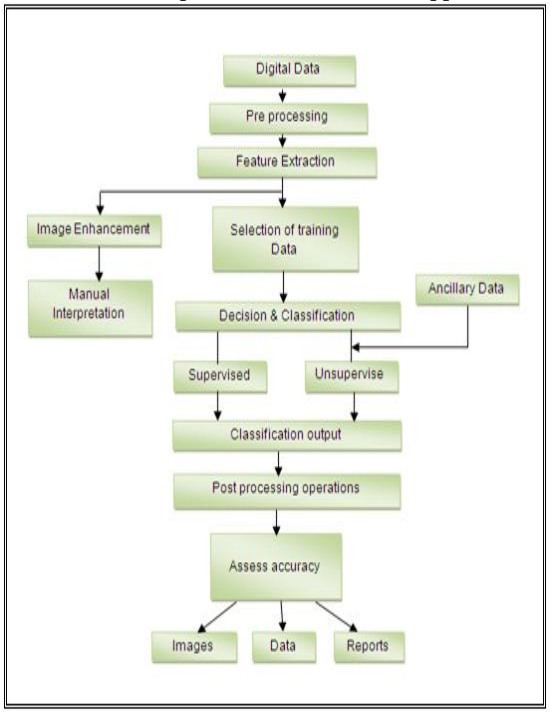


Figure1 Block Diagram of Digital image processing

Main steps used in digital image processing are as shown in Figure 1.

The NL-means algorithm tries to take advantage of the high degree of redundancy of any natural image. By this, we simply mean that every small window in a natural image has many similar windows in the same image. The NL-means algorithm chooses for each pixel a different average configuration adapted to the image. For a given pixel *i*, we take into account the similarity between the neighborhood configuration of *i* and all the pixels of the image. The similarity between pixels is measured as a decreasing function of the Euclidean distance of the similarity windows.

Non-local means filter uses all the possible self-predictions and self-similarities the image can provide to determine the pixel weights for filtering the noisy image, with the assumption that the image contains an extensive amount of self-similarity. As the pixels are highly correlated and the noise is typically independently and identically distributed, averaging of these pixels results in noise suppression thereby yielding a pixel that is similar to its original value. The non-local means filter removes the noise and cleans the edges without losing too many fine structure and details. But as the noise increases, the performance of non-local means filter deteriorates and the denoised image suffers from blurring and loss of image details. This is because the similar local patches used to find the pixel weights contain noisy pixels. The NL-means algorithm is easily extended to the denoising of image sequences and video. The denoising algorithm involves indiscriminately pixels not belonging only to the same frame but also to all frames in the image. The algorithm favors pixels with a similar local configuration, as the similar configurations move, so do the weights. Thus, the algorithm is able to follow the similar configurations when they move without any explicit motion computation.

An improved nonlocal means denoising algorithm method is quite intuitive and potentially very powerful, the PSNR and visual results are somewhat inferior to other recent state-of-the-art non-local algorithms, like KSVD and BM-3D. In this paper, we have shown that the NL means algorithm is basically the first iteration of the Jacobi optimization algorithm for robustly estimating the noise-free image using Matlab.

The moments (geometric moments, complex moments) and moment-based invariants with respect to various image degradations and distortions (rotation, scaling, affine transform, image blurring, etc.) which can be used as shape descriptors for classification. We explain a general theory how to construct these invariants and show also a few of them in explicit forms.

The search for efficient image denoising methods is still a valid challenge at the crossing of functional analysis and statistics.

II. SYSTEM ANALYSIS

The main idea is to replace each pixel with a weighted average of other pixels with similar neighbourhoods. The main difference between NLM and previous approaches is that the weights in the NLM filter do not depend on the spatial distance between target patches and candidates but depend on the difference of intensity values.

Approaches addressing improving NLM can be categorized as: Acceleration and Denoising Performance Improvement.

. Approaches addressing improving NLM can be categorized as acceleration and denoising performance improvement. But in this paper we improve the non local means by using moment invariants in pre-selection and RIBM in filtering process. This gives more reliable clustering results due to the 'invariant under noise' characteristic of Hu's moment invariants. On this basis, RIBM provides the rotation invariant weight calculation within each cluster. The experimental results show that this method outperforms the original NLM in terms of both quantitatively and visual quality.

III. EXISTING SYSTEM

The non-local means algorithm does not make the same assumptions about the image as other methods. Instead it assumes the image contains an extensive amount of self-similarity. Efros and Leung originally developed the concept of self-similarity for

texture synthesis. An example of self-similarity is displayed in Figure 2. The figure shows three pixels *p, q1,* and *q2* and their respective neighborhoods. The neighborhoods of pixels *p* and *q1* are similar, but the neighborhoods of pixels *p* and *q2* are not similar. Adjacent pixels tend to have similar neighborhoods, but non-adjacent pixels will also have similar neighborhoods when there is structure in the image. Pixels *p* and *q1* have similar neighborhoods, but pixels *p* and *q2* do not have similar neighborhoods. Because of this, pixel *q1* will have a stronger influence on the denoised value of *p* than *q2*.

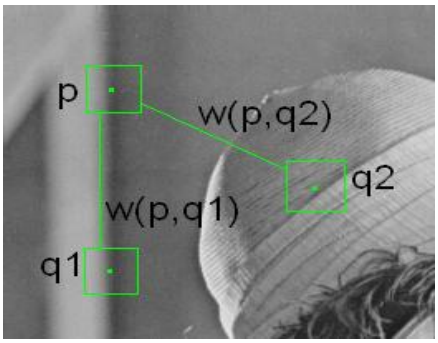


Figure 2 Example of self-similarity in an image

IV. PROPOSED SYSTEM

We propose a system is used to produce a denoised image in which the noise has been removed and most of the details are retained. Gaussian blur provides the pre-processing for pre-classification. In the original NLM, there is no pre-processing step. K-means clustering on moment invariants of the blurred noisy image serves as the pre-classification for our filtering process. In the original NLM, all target patches have fixed candidate sets, which is either the whole image or the neighborhood centered at them. RIBM is calculated on the input noisy patches which have been clustered by using a look-up table (LUT) from Step .This step introduces a new similarity term for nonlocal filtering. In the original NLM, the similarity term just relates to the Euclidean distance. Advantages of proposed system are:

- Gaussian blurred image provides better classification before weighted averaging.
- In addition, RIBM adds more "similar patches" which have been rotated by certain angles to make them more correlated to the reference patch.
- Less time consumption and high quality.

V. RELATED WORK

Given a noisy image $v = \{ v(i) | i \in \Omega \}$, $\Omega \subset R^2$, the restored intensity of the pixel $NL(v)(i)$, is a weighted average of all intensity values within the neighbourhood .

$$NL(v)(i) = \sum_{j \in I} w(i,j)v(j) \qquad (1)$$

where v is the intensity function, $v(j)$ is the intensity at pixel j , and $w(i, j)$ is the weight assigned to $v(j)$ in the restoration of pixel . The weights can be calculated by [7].

$$w(i,j) = \frac{1}{Z(i)} e^{-\|v(Ni)-v(Nj)\|^2_{2,a}/h^2} \qquad (2)$$

where Ni denotes a patch of fixed size and it is centered at the pixel . The similarity is measured as a decreasing function of the weighted Euclidean distance. a>0 is the standard deviation of the Gaussian kernel, is the normalization constant with $Z(i) = \sum_j w(i, j)$, and acts as a filtering parameter.

To find a set of reliable candidates that are similar to the current patch from a whole image, two categories of methods are applied: 1) pre-classification and 2) defining new similarity terms.

Differences between our approach and NLM are:

1. Gaussian blur provides the pre-processing for pre-classification. In the original NLM, there is no pre-processing step.
2. K -means clustering on moment invariants of the blurred noisy image serves as the pre-classification for our filtering process. In the original NLM, all target patches have fixed candidate sets, which is either the whole image or the neighbourhood centred at them.
3. RIBM is calculated on the input noisy patches which have been clustered by using a look-up table (LUT) from Step 2. This step introduces a new similarity term for nonlocal filtering. The calculation of weights is as explained. In the original NLM, the similarity term just relates to the Euclidean distance.

VI. METHODS

K-Means Clustering: Adaptively classify the acquired data by choosing appropriate centroid. This algorithm partitions into K classes while minimizing the within-cluster sum of squares where the mean of the centroids showing maximum cluster similarity is is fixed as standard centroids as shown in Figure 3.

Rotationally Invariant Block Matching (RIBM): The central problem is to estimate the angle of rotation between two corresponding blocks. This is done with the help of one point correspondence: we estimate the angle by which a certain pixel in the block called centroid is rotated around the blocks center. Natural requirements at such a centroid are that it is robust under noise and easy to calculate. Furthermore, we identify all points within a block by vectors pointing from its center to the points coordinates. We can then describe the basic idea of rotationally invariant block matching (RIBM) in a simple generic algorithm: 1. Estimate the angle of rotation between the blocks.2. To each pixel in the first block, find the position of the corresponding pixel in the second block by rotating its vector by this estimated angle.

The summed distances represent the total distance of the two blocks. It obviously makes sense to use circles as blocks here. This algorithm can be extended to detect not only rotated, but also mirrored versions of the reference block. If we have found a mirrored version, we can again mirror it at an arbitrary axis and then apply the algorithm.

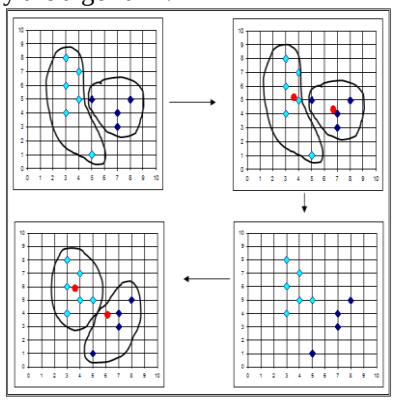


Figure 3 K-means clustering algorithm

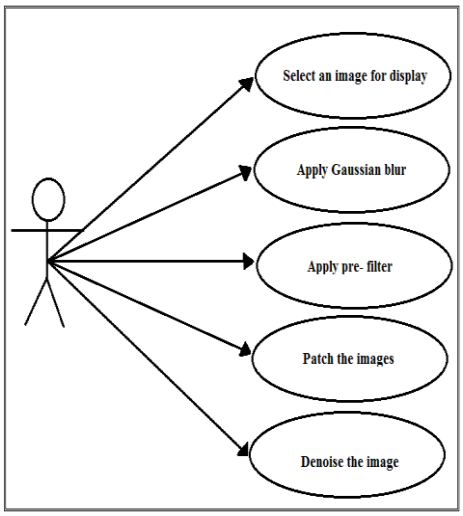


Figure 4 USE CASE DIAGRAM

VII. SAMPLE CODING

INPUT AN IMAGE:

```
% --- Executes on button press in Inputimage.
function Inputimage_Callback(hObject, eventdata, handles)
% hObject    handle to Inputimage (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
global filename
global pathname
global image
[filename pathname]=uigetfile('*.tif','Select An Image');
image=imread([pathname filename]);
axes(handles.axes1);
imshow(image);
axis equal;axis off;
```

PRE-FILTERING:

```
% --- Executes on button press in Pre_Filtering
function Pre_Filtering_Callback(hObject, eventdata, handles)
% hObject    handle to Pre_Filtering (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
global noiimage filimage
gimage=fspecial('gaussian',[5 5],2);
filimage=imfilter(noiimage,gimage,'same');
axes(handles.axes1);
imshow(filimage);axis equal;axis off;
```

PATCH SEPARATION:

```
% --- Executes on button press in PatchSeparation.
function PatchSeparation_Callback(hObject, eventdata, handles)
% hObject    handle to PatchSeparation (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
global filimage
[m n] = size(filimage);
k = 1;
w = 15;
imagepatch=cell(225,1);
for i = 1:w:m
   for j = 1:w:n
      temp = filimage(i:i+w-1,j:j+w-1);
      figure(1);
      subplot(15,15,k),imshow(temp);axis off;
      imagepatch{k,1}=temp(:,:);
      k = k+1;
   end
end
save imagepatch imagepatch
```

CLASSIFICATION and DENOISING:

```
% --- Executes on button press in Classification_Denoising
function Classification_Denoising_Callback(hObject, eventdata, handles)
% hObject    handle to Classification_Denoising (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
global image
load trained
load noisemomentfea
load imagepatch
load noiimpatch
classification=classify(noisemomentfea,momentfea,IDX)
save classification classification
% ------------------Denoising-----------------------
```

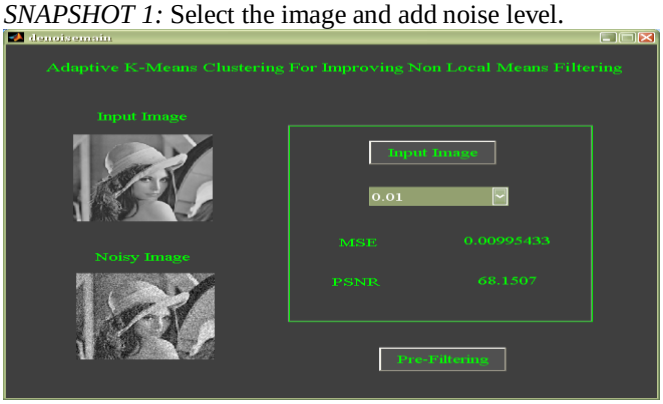
```
A=cell(225,1);
rr=size(imagepatch,1);
testclassification=zeros(225,1);
for kv=1:rr

testnoisepatch(kv,1:7)=invmoments(noiimpatch{kv,1}
);

testclassification=classify(testnoisepatch,momentfea,I
DX);
      if testclassification(kv,1)==1
         ind1=find(IDX(:,1)==1);
          for ooi=1:numel(ind1)
            oripatch=imagepatch{ooi,1};
            filpatch=noiimpatch{kv,1};
            similarity1(ooi,1)=corr2(oripatch,filpatch);
          end
            simpatchind=max(similarity1(:,1));
            insim=find(similarity1(:,1)==simpatchind);
            image1=imagepatch{insim,1};
            image2=filpatch;
             A{kv,1} = nlmfil(image2,image1);
            %%%%%%%%%%%%%%
      end
      if testclassification(kv,1)==2
         ind2=find(IDX(:,1)==2);
          for ooii=1:numel(ind2)
            oripatch=imagepatch{ooii,1};
            filpatch=noiimpatch{kv,1};
            similarity2(ooii,1)=corr2(oripatch,filpatch);
          end
            simpatchind=max(similarity2(:,1));
            insim=find(similarity2(:,1)==simpatchind);
            image1=imagepatch{insim,1};
            image2=filpatch;
            %%%%%%%%%%%%%%
            A{kv,1} = nlmfil(image2,image1);
          %%%%%%%%%%%%%
      end
      if testclassification(kv,1)==3
         ind3=find(IDX(:,1)==3);
          for oooi=1:numel(ind3)
            oripatch=imagepatch{oooi,1};
            filpatch=noiimpatch{kv,1};
            similarity3(oooi,1)=corr2(oripatch,filpatch);
          end
            simpatchind=max(similarity3(:,1));
            insim=find(similarity3(:,1)==simpatchind);
            image1=imagepatch{insim,1};
            image2=filpatch;
            %%%%%%%%%%%%%%
             A{kv,1} = nlmfil(image2,image1);
      end
end
j=1;
for i=1:225
   figure(3);subplot(15,15,j);
   imshow(A{i,1});
   axis off;
   j=j+1;
end
%----- Reconstruction ----------
imrecons = zeros(225,225);
s = 1;t=15;
for i = 1:15:225
   for j = 1:15:225
        imrecons(i:i+t-1,j:j+t-1) = A{s,1};
      s = s+1;
   end
end
 figure(4),
 imshow(imrecons);
 title('DENOISED IMAGE');
 [MSE1,                PSNR1]                =
Calc_MSE_PSNR(image,imrecons)
set(handles.text4,'String',MSE1);
set(handles.text6,'String',PSNR1);
```

CALCULATION OF MEAN SQUARE ERROR (MSE) AND PEAK SIGNAL TO NOISE RATIO (PSNR):

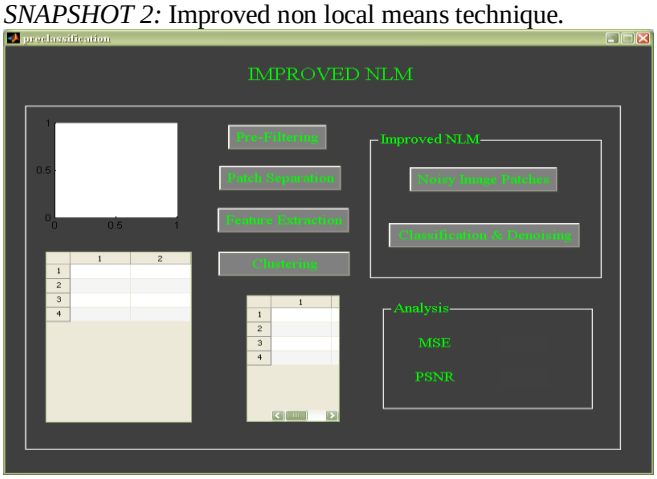
```
%---Function to calculate mse and psnr
function          [MSE,          PSNR]          =
Calc_MSE_PSNR(clean,denoised)
N = prod(size(clean));
clean      =      double(clean(:));      denoised      =
double(denoised(:));
t1 = sum((clean-denoised).^2); t2 = sum(clean.^2);
MSE = t1/N;
PSNR = 10*log10(255*255/MSE);
```

VIII. RESULTS AND DISCUSSIONS

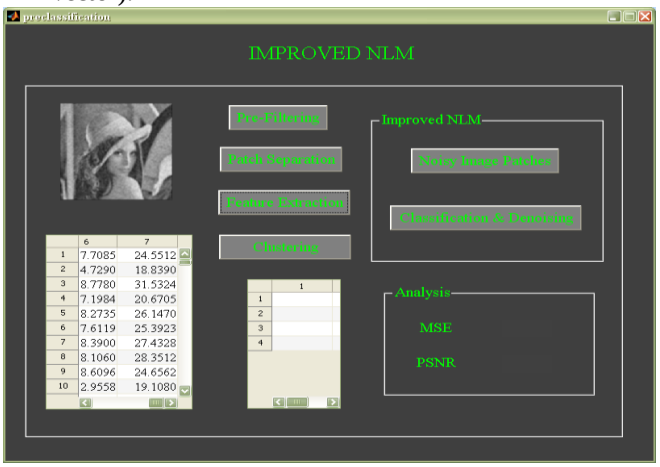
SNAPSHOT 1: Select the image and add noise level.



SNAPSHOT 2: Improved non local means technique.

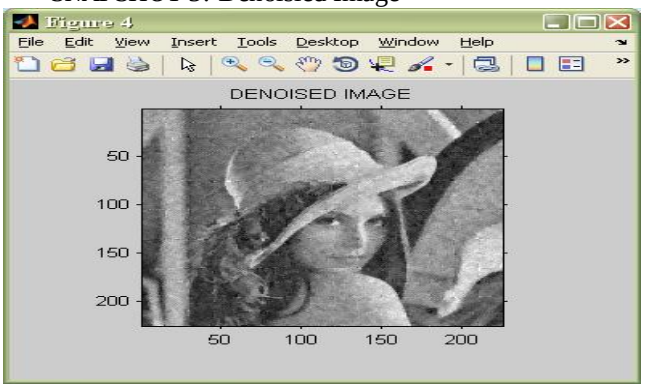


SNAPSHOT 3: Feature Extraction: The process of transforming the input data into a reduced representation set of features (also named features vector).

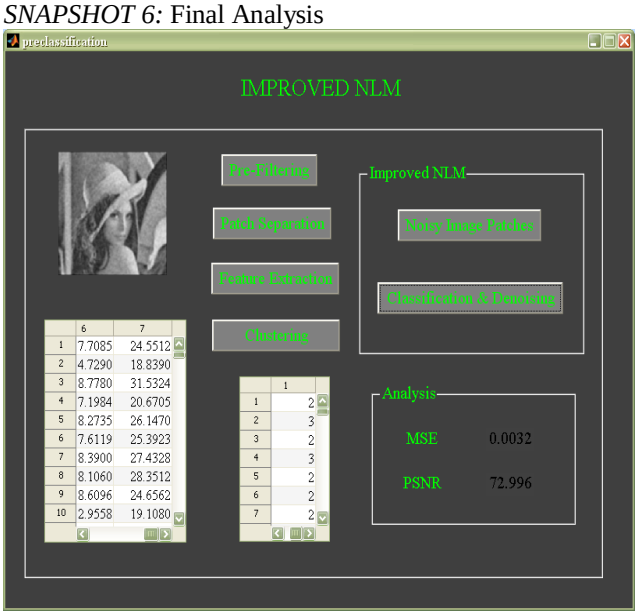


SNAPSHOT 4: Clustering: The process of adaptively classifying the acquired data by choosing appropriate centroid.

SNAPSHOT 5: Denoisied Image



SNAPSHOT 6: Final Analysis



IX. ACKNOWLEDGMENT

We would like to thank pooja, Rangathan, Rathan and Sowmyashree for their work and support.

REFERENCES

[1] L. Shao, H. Zhang, and G. de Haan, "An overview and performance evaluation of classification-based least squares trained filters,"IEEE Trans. Image Process.,vol. 17, pp. 1772–1782, Oct. 2008.

[2] M. Protter and M. Elad, "Image sequence denoising via sparse and redundant representations,"IEEE Trans. Image Process., vol. 18, pp. 27–35, Nov. 2003.

[3] G. Varghese and W. Zhou, "Video denoising based on a spatiotem-poral Gaussian scale mixture model,"IEEE Trans. Circuits Syst. Video Technol., vol. 20, no. 7, pp. 1032–1040, Jul. 2010.

[4] F. Luisier, T. Blu, and M. Unser, "SURE-LET for orthonormal wavelet-domain video denoising,"IEEE Trans. Circuits Syst. Video Technol., vol. 20, no. 6, pp. 913–919, Jun. 2010.

[5] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian, "Image denoising by sparse 3-D transform-domain collaborative filtering,"IEEE Trans. Image Process., vol.16, pp. 2080–2095, Aug. 2007.

[6] A. Buades, B. Coll, and J. M. Morel, "A review of image denoising algorithm, with a new one,"Simulation, vol. 4, pp. 490–530, 2005.

[7] M. Mahmoudi and G. Sapiro, "Fast image and video denoising via non-local means of similar neighborhoods,"IEEE Signal Process. Lett., vol. 12, pp. 839 -842, Dec. 2005.

[8] P. Coupe, P. Yger, S. Prima, P. Hellier, C. Kervrann, and C. Barillot, "An optimized blockwise nonlocal means denoising filter for 3-D mag-netic resonance images,"IEEE Trans. Med. Imag., vol. 27, no. 4, pp. 425–441, Apr. 2008.

[9] T. Thaipanich, O. B. Tae, W. Ping-Hao, X. Daru, and C. C. J. Kuo, "Improved image denoising with adaptive nonlocal means (ANL-means) algorithm," IEEE Trans. Consum. Electron., vol. 56, no. 4, pp. 2623–2630, Nov. 2010.

[10] J. Wang, Y. Guo, Y. Ying, Y. Liu, and Q. Peng, "Fast non-local algo-rithm for image denoising," inProc. IEEE Int. Conf. Image Process, Atlanta, GA, USA, 2006, pp. 1429–1432.

[11] B. Goossens, H. Luong, A. Pizurica, and W. Philips, "An improved non-local denoising algorithm," inProc. Int. Workshop on Local and Non-local Approximation in Image Process, Tuusalu, Finland, 2008, pp. 143–156.

[12] P. Chao, O. C. Au, D. Jingjing, Y. Wen, and Z. Feng, "A fast NL-means method in image denoising based on the similarity of spatially sampled pixels," inProc. IEEE Int. Workshop on Multimedia Signal Processing, Rio de Janeiro, Brazil, 2009.

[13] D. Tschumperle and L. Brun, "Non-local image smoothing by applying anisotropic diffusion PDE's in the space of patches," inProc. IEEE Int. Conf. Image Process., Cairo, Egypt, 2009, pp. 2957–2960.

[14] Ruomei Yan, Ling Shao, , Sascha D. Cvetkovic´, and Jan Klijn , "Improved Nonlocal Means Based on Pre-Classification and Invariant Block Matching," in journal of display technology, Vol. 8, No. 4, April 2012, PP. 212-218.