REFERENCES

[1] R. O. K. Reddy, B. E. Reddy and E. K. Reddy, "Classifying the similarity and defect fabric textures based on gray level co-occurrence matrices and binary pattern schemes," *International Journal of Information Engineering and Electronic Business*, vol. 5, pp. 25-33, Nov. 2013.

[2] A. L. Amet, A. Ertüzün and A. Erçil, "An efficient method for texture defect detection: subband domain co-occurrence matrices," *Image and Vision Computing*, vol. 18, pp. 543-553, May 2000.

[3] J. Kyllönen and M. Pietikäinen, "Visual inspection of the parquet slabs by combining color and texture," in *proc. of the IAPR Workshop on Machine Vision Applications*, Tokyo, Japan, 2000, pp. 187-192.

[4] I. Novak and Z. Hocenski, "Texture feature extraction for a visual inspection of ceramic tiles," in *proc. of the IEEE International Symposium on Industrial Electronics*, Dubrovnik, Croatia, 2005, pp. 1279-1283.

[5] A. Kumar, "Neural network based detection of local textile defects," *Pattern Recognition*, vol. 36, pp. 1645-1659, July 2003.

[6] R .Stojanovic, P .Mitropulos, C .Koulamas, S .Koubias, G .Popoulos and G .Karayanis, "Automated detection and neural classification of local defects in textile web," in *Proc. of the Int. Conf. on Image Processing And Its Applications*, Manchester, United Kingdom, 1999, pp. 647-651.

[7] C.J. Kuo and C. Lee, "A back-propagation neural network for recognizing fabric defects" *Textile Research Journal*, vol. 73, pp. 147-151, Feb. 2003.

[8] S. H. Jeong, H. T. Choi, S. R. Kim, J. Y. Jaunng and S. H. Kim, "Detecting fabric defects with computer vision and fuzzy rule generation part II: Defect identification by a fuzzy expert system, "*Textile Research Journal*, vol. 71, pp. 561-573, June 2001.

[9] J. Chang, G. Han, J.M. Valverde, N. C. Griswold,J. F. Duque-Carrilloand E. S.´anchez-Sinencio," Cork quality classification system using a unified image processing and fuzzy-neural network methodology," *IEEE Transactions on Neural Networks*, vol. 8, pp. 964-974, July 1997.

[10] M. Niskanen, O. Silvén, and H. Kauppinen, "Color and texture based wood inspection with non-supervised clustering," *Machine Vision and Applications*, vol. 13, pp. 275-285, March 2003.

[11] H. Kauppinen, H. Rautio and O. Silvén, "Non-segmenting defect detection and SOM based classification for surface inspection using color vision," in *proc. of the 1999 int. conf. on Polarization and Color Techniques in Industrial Inspection,* Munich, Germany, 1999.

[12] J. Tian, M. H. Azarian, and M. Pecht, "Anomaly detection using self-organizing maps based k-nearest neighbour algorithm," in *Proc. of the Int. Conf. of the Prognostics and Health Management Society,* 2014.

[13] A.Monadjemi, "*Towards Efficient Texture Classification*," Ph.D. dissertation, University of Bristol, Bristol, 2004.

[14] J. Iivarinenand A. Visa, "An adaptive texture and shape based defect classification," in *Proc. of the Int. Conf. on Pattern Recognition,* Brisbane, Australia, 1999, pp.117-122.

[15] A. Kumar, "Computer vision-based fabric defect detection: a survey," *IEEE Transactions on Industrial Electronic*s, vol. 55, pp. 348 - 363, Jan 2008.

[16] X. Xie, "A review of recent advances in surface defect detection using texture analysis techniques," *Electronic Letters on Computer Vision and Image Analysis*, vol. 7, pp. 1-22, June 2008.

[17] M. Vapola, O. Simula, T. Kohonen and P. Meriläinen, "Representation and identification of fault conditions of an anaesthesia system by means of the self-organizing map," in *Proc. of the Int. conf. on Artificial Neural Networks,* Sorrento, Italy, 1994.

[18] T. E. Mursalin, F. Z. Eishita and A. R. Islam, "Fabric defect inspection system using neural network and microcontroller," *Journal of Theoretical and Applied Information Technology*, vol. 19, pp. 560-570, June 2006.

[19] G. M. Nasira1 and P.Banumathi, "Automatic defect detection algorithm for woven fabric using artificial neural network techniques," *International Journal of Innovative Research in Computer and Communication Engineering*, vol. 2, pp. 2620-2624, Jan. 2014.

[20] L. A. O. Martins, F. L.C. P´adua and P. E. M. Almeida, "Automatic detection of surface defects on steel using computer vision and artificial neural networks," in *Proc. of the int. conf. on IEEE Industrial Electronics Society*, Glendale, USA, 2010, pp. 1081-1086.

[21] L. Liquing, T. Gia and X. Chen, "Automatic recognition of the fabric structure based on digital image," *Indian Journal of Textile research*, vol. 33, pp. 388-391, Dec. 2008.

[22] P.F. Li, J. Wang, H.H. Zhang and J.F. Jing, "Automatic woven fabric classification based on support vector machine," in *proc. of the Conf. on Automatic Control and Artificial Intelligence*, Xiamen, China, 2012.

[23] K. Shiranitaa, T. Miyajimab and R. Takiyamac, "Determination of the meat quality by texture analysis," *Pattern Recognition Letters*, vol. 19, pp. 1319-1324, Dec. 1998.

[24] M. S. Hitam, M. Y. H. Muraslan, M. M. Deris and M. Y. M. Saman, "Image texture classification using gray level co-occurrence matrix and neural network," *Europen journal of sciencetific Research*, vol. 75, pp. 591-597, Nov. 2005.

[25] C. Feng ,J. Kuo and C. Tsai, "Automatic recognition of fabric nature by using the approach of texture analysis" *Textile Research Journal*, vol. 76, pp. 564-579, May 2006.

[26] R. M. Haralick and K. Shanmugam, "Textural features for the image classification" *IEEE Transactions on Systems, Man, and Cybernetics*, vol. SMC-3, pp. 610-621, Nov. 1973.

[27] A. S. Malek, "*Online fabric inspection by image processing technology*," Ph.D Thesis, Doctoral School, University of Haute Alsace, Haute Alsace, 2012.

[28] l. soh and c. tsatsoulis, "Texture analysis of SAR sea ice imagery using gray level co-occurrence matrices," *IEEE transactions on geoscience and remote sensing*, vol. 37, pp. 780-795, Mar. 1999.

[29] R.C. Gonzalez and R.E. Woods, *Digital Image Processing*, 3rd ed. New Jercy: Prentice-Hall, 2006.

[30] T. Ojala, M. Pietikäinen and T. Mäenpää, "Multiresolution gray scale and the rotation invariant texture classification with local binary patterns" *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, pp. 971-987, Jul. 2002.

[31] T. Kohonen, "Essentials of the self-organizing map," *Neural Networks*, vol. 37, pp. 52-65, Jan. 2013.

[32] C.M. Bishop, *Pattern Recognition and Machine Learning*, Berlin:Sringer, 2006.

APPENDIX A MATLAB SCRIPTS

A.1 Pre-processing of images

```
rgbImage = imread(fullFileName);
greenChannel = rgbImage(:, :, 2);
GG=greenChannel;
```

Green channel

```
I=double(GG)/255;
G=power(I,5);
```

Gama correction

```
im=im2double(G);
fftlogim=fft2(log(im+0.01));
f=butterhp(im,10,1);
c=fftlogim.*f;
h=real(ifft2(c));
h1=exp(h);
f1=abs(h1);
fm=max(f1(:));
homo=(f1/fm);G=homo;
```

Homomorphic filter

```
stE1 = strel('disk',25);
Top=imtophat(homo,stE1);
Bot=imbothat(homo,stE1);
EnIm = (K+Top)-Bot;
```

Morphological gradient

```
hist = adapthisteq(EnIm);
```

Histogram equalization

```
function [ out ] = butterhp(im,d,n)
h=size(im,1);
w=size(im,2);
[x,y]=meshgrid(-floor(w/2):floor(w-1)/2, -floor(h/2):floor(h-1)/2);
out =1./(1.+(d./(x.^2+y.^2).^0.5).^(2*n));
end
```

Butterworth High-pass

A.2 Texture extraction

Loop for splitting the image:

```
t1 = 100; t2 = 150;
A1 = zeros(1, (600/t1)-1);
A2 = zeros(1, (900/t2)-1);

for k1 = 0 : (600/t1)-1
    A1(k1+1) = t1*k1+1;
end

for k2 = 0 : (900/t2)-1
    A2(k2+1) = t2*k2+1;
end

va=zeros(t1,t2);n=0;

for x=1:600
    for y=1:900
        if (any(x==A1) && any(y==A2))

          %feature extraction from window

        end
    end
     waitbar(i/600,2);
end
```

Local texture model (LBP):

```
%inside spliting loop
n=n+1;
bTemp = b(i:i+t1-1, j:j+t2-1);
nFiltSize=32;nFiltRadius=16;
filtR=generateRadialFilterLBP(nFiltSize, nFiltRadius);
val = efficientLBP(d, 'filtR', filtR, 'isRotInv', false,  'isChanWiseRot', false);
va = efficientLBP(bTemp, 'filtR', filtR, 'isRotInv', true,  'isChanWiseRot', false);
vall{n}=mat2cell(va);
```

Extraction of histogram of local texture model:

```
for xx=1:28 %Fabric type identification training loop

 vall{n}=mat2cell(va);% LBP cell array of image windows

for ii=1:(600/t1)*(900/t2)

hitb=cast(cell2mat(vall{ii}),'double');% cell>mat & casting to double / ii th half of
a2 = unique(hitb);%unique elements in hit b
nb = [a2,histc(hitb(:),a2)];%histogram based on unique elements
nb(1,:)=[];%empty first colum
qa = [1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,3
aInd = ismember(qa,nb(:,1));% binery vector indicate qa element is in nb
xf = zeros(length(qa),2);
xf(:,1) = qa;% 1 st row of xf = qa
xf(aInd,2) = nb(:,2); % add respective count to 2 nd row of xf based on 1 & 0 of nb
xf(not(aInd),2) = NaN;% NAN to nb's 0
xf(isnan(xf))=0;% 0 to xf's NAN
KKK=xf(:,2);% 2nd coloum of xf
vec{ii}=KKK;% cell array of histogram of ii th split

end
```

Development of GLCM and statistical measurements:

```
%in side splliting loop
n=n+1;
bTemp1 = b1(x:x+t1-1, y:y+t2-1);
glcm2 = graycomatrix(bTemp1,'NumLevels',16,'offset', [0 8], 'Symmetric', true);
out1 = GLCM_features(glcm2,0);

v13=out1.autoc; v14=out1.contr; v15=out1.corrm;v16=out1.corrp; v17=out1.cprom;v
v19=out1.dissi; v110=out1.entro; v111=out1.homom; v112=out1.homop; v113=out1.ma
v115=out1.savgh; v116=out1.svarh; v117=out1.senth; v118=out1.dvarh; v119=out1.d

vat =[v114 v115 v116 v117 v16 v14 v110 v17 v18 v19];
vallt{n}=mat2cell(vat);
```

A.3 Development of KSOM

Surface type wise data base for KSOM training:

```
fvec = (cat(2, vec{:}))'; % vector of all ii parts of image
XX1{xx} = fvec; % cell array of  xx th image

switch xx % Condition to selct image surface type from image folder
   case num2cell(1:4)
        a=1;
   case num2cell(5:8)
        a=2;
   case num2cell(9:12)
        a=3;
   case num2cell(13:16)
        a=4;
   case num2cell(17:20)
        a=5;
   case num2cell(21:24)
        a=6;
   case num2cell(25:28)
        a=7;
end

fvecc = [a.*(ones(size(fvec,1),1)) fvec]; % add additional row(a value
XXX{xx} = fvecc;% put fvecc to xx th cell
waitbar(xx/28,hf2,sprintf('%1d''of 28',xx))
end
```

Development and training of KSOM:

```
feat=vertcat(XX1{:});% all xx cells to one matrix (xx*ii halfs)
feature = feat(randperm(size(feat,1)),:); % randomize vectors coloum vise
X=feature;
N=20;
lattice = 'hexa';
neigh = 'gaussian';
radius_coarse = [6 .5];
trainlen_coarse = 30000;
radius_fine = [.5 .5];
trainlen_fine = 1000;
smI = som_lininit(X,'msize',[N N],'lattice',lattice,'shape','sheet');
smC = som_batchtrain(smI,X,'radius',radius_coarse,'trainlen',trainlen_coarse, 'ne
sm = som_batchtrain(smC,X,'radius',radius_fine,'trainlen',trainlen_fine, 'neigh',
M = sm.codebook;
norms2 = sum(M.*M,2);
save('Mc.mat','M');
save('norms2c.mat','norms2');
```

Data base generation for labeling of KSOM:

```
featuree=vertcat(XXX{:});
Xc=featuree; % vector with fabric type
B = repmat(featuree(:,1),1,(900/t2)*(600/t1));%type matrix
Xc(:,1) = []; % empty fabric type detail
hits = zeros(400,7);

for u=1:((600/t1)*(900/t2))*xx;

X1 = Xc(u,:)';
Y(:,u) = norms2 - 2*M*X1;
[YY,YYY]=sort(Y);

for i=1:3
    cc=YYY(i);% Nearest 3 naighbours(3 BMU)
switch B(u,i)% put to bins based on type and BMU
   case 1
        hits(cc,1) = hits(cc,1) + 1;
   case 2
        hits(cc,2) = hits(cc,2) + 1;
   case 3
        hits(cc,3) = hits(cc,3) + 1;
   case 4
        hits(cc,4) = hits(cc,4) + 1;
   case 5
        hits(cc,5) = hits(cc,5) + 1;
  case 6
        hits(cc,6) = hits(cc,6) + 1;
  case 7
        hits(cc,7) = hits(cc,7) + 1;
end
end
end
```

3-NN identification

Utilizing 3-NN for Labeling of KSOM

Labeling and coloring of KSOM:

```
colormapigray =  colormap('gray');
for i = 1:7;subplot(1,2,1);
hc1=som_cplane(sm, hits(:, i));
set(hc1,'edgecolor','none');
pause(3);
end

nodelabels = zeros(400,1);
for i=1:length(nodelabels);
[C,c] = max(hits(i,:));
nodelabels(i) = c;
end

colormapigray2 =  colormap('hsv');
subaxis(1,2,1, 'Spacingvert', 1, 'Padding', 0, 'Margin', 0);
hc2=som_cplane(sm, nodelabels);
set(hc2,'edgecolor','none');
```

Calculation of 'class belongingness':

```
out=sum(hits,2);
mem = bsxfun(@rdivide, hits, out);
mem(out==0) = 0;
mem(hits==0) = 0;
save('memc.mat','mem');
```

Execution of KSOM and integration of clustering decisions:

```
fvec = (cat(2, vec{:}))'; % vector of all ii parts of image
X=fvec;
decm=zeros(36,13);

for  u=1:36;
X1 = X(u,:)';
Y = norms2a - 2*Ma*X1;
[C,c] = min(Y);
decm(u,:)=mem(c,:);
end

[su,deci]=max(sum(decm,1));
fprintf('fabric is type %d .\n',deci);
```

Associated processes during training of KSOM:

```
featuree=vertcat(XXX{:});
Xc=featuree; % vector with fabric type
B = repmat(featuree(:,1),1,(900/t2)*(600/t1));
Xc(:,1) = []; % empty fabric type detail

for uu=1:((600/t1)*(900/t2))*xx;
X11=Xc(uu,:)';
d1(uu,:)=norms2 - 2*M*X11;
end
d2=d1';
[BB,II] = sort(d2);
DD=BB((1:3),1:xx*(900/t2)*(600/t1));
DI=II((1:3),1:xx*(900/t2)*(600/t1));
E=sum(DD);

index = zeros(xx*(900/t2)*(600/t1),7);

for u=1:((600/t1)*(900/t2))*xx;

switch featuree(u,1)
   case 1
       index(u,1)=E(u);
   case 2
       index(u,2)=E(u);
   case 3
       index(u,3)=E(u);
   case 4
       index(u,4)=E(u);
   case 5
       index(u,5)=E(u);
   case 6
       index(u,6)=E(u);
   case 7
       index(u,7)=E(u);
end

end

save('indexc.mat','index');
```

Calculation of *'anomaly indicator'*

Storing the *anomaly indicators* results during training in database

A.4 Detection of Defects using KSOM

Using the KSOM for detection of defects:

```
X = (cat(2, vec{:}))'; % vector of all ii parts of image

for uu=1:((600/t1)*(900/t2));
X11=X(uu,:)';
d1=norms2 - 2*M*X11;
d2(uu,:)=d1;
end
d3=d2';
[BB,II] = sort(d3);
DD=BB((1:3),1:(900/t2)*(600/t1));
DI=II((1:3),1:(900/t2)*(600/t1));
Edd=sum(DD);
```

Calculation of *anomaly indicators* w.r.t. all the Sub-images

```
max1=max(index(:,1));
min1=min(index(:,1));
std1=nanstd(index(:,1));
thre = (Edd >max1-3*std1);
```

Thresholding process

Development of binary mask into binary decision:

```
re=(reshape(thre,[(900/t2),(600/t1)]))';
fdeci = sum(re(:));

imshow2 = reshape(repmat(reshape(re',1,[]),t2,1),[],size(re,1))';
imshow3 = reshape(repmat(imshow2(:)',t1,1),[],size(imshow2,2));

maskedimage1 = b1;
maskedimage1(~imshow3) = 0;

Dedge=imshow3;
s=strel('disk',8,0);
Fedge=imerode(Dedge,s);
imshow4=1-(Dedge-Fedge);
rgbImage =imread(fullFileName);
jkm=imresize(rgb2gray(rgbImage),[600 900]);
maskedimage2 = jkm;
maskedimage2(~imshow4) = 0;
```

A.5 Development of graphical user interfaces

GUI for execution of input images:

```
M = struct2array(load('M4.mat'));
norms2 = struct2array(load('norms24.mat'));
index=struct2array(load('index4.mat'));

DlgH = figure(1);
set(1,'units','normalized','outerposition',[0 0 1 1]),set(1,'ToolBar','none');
set(1,'MenuBar','none');
set(1,'Name',sprintf('Fabric defects detection'),'NumberTitle','off');

obj1 = uicontrol('Style', 'PushButton','String', 'Exit now','Callback', 'delete(gcbf)',
'FontSize',11.5,'FontWeight','bold');

p1 = uipanel(DlgH,'Title','Input directory panel','Position',[.015 .81 .4 .12],'FontSi
,'BackgroundColor',[255/255 153/255 255/255]);

p2 = uipanel(DlgH,'Title','Input panel','Position',[.015 .07 .4 .12],'FontSize',11,'Fo
[255/255 153/255 255/255]);

p3 = uipanel(DlgH,'Title','Decision','Position',[.58 .81 .4 .12],'FontSize',11,'FontWe
[255/255 153/255 255/255]);

p4 = uipanel(DlgH,'Title','Stage of Processing','Position',[.58 .07 .4 .12],'FontSize',
[255/255 153/255 255/255]);

textH1 = uicontrol(1,'Style','edit','String','C:\Users\dimuthu\Desktop\type4','Positio

textH2 = uicontrol(1,'Style','edit','String','t.png','Position',[40 90 290 20],'FontSi

obj2 = uicontrol('Position',[340 80 170 40],'String','Continue','Callback','uiresume(g

uiwait(gcf);

myFolder = strcat( get(textH1, 'string')) ;
jpgFileName = strcat( get(textH2, 'string'));
fullFileName = fullfile(myFolder, jpgFileName);
```

GUI for extraction of features:

```
myFolder = 'C:\Users\dimuthu\Desktop\images';
DlgH2 = figure(2);
set(2,'units','normalized','outerposition',[0 0 1 1]),set(2,'ToolBar','none');
set(2,'MenuBar','none');
set(2,'Name',sprintf('Feature Extraction'),'NumberTitle','off');

objdel1 = uicontrol('Style', 'PushButton','String', 'Exit now','Callback', 'dele
'FontSize',11.5,'FontWeight','bold');

p11 = uipanel(DlgH2,'Title','Filtering','Position',[.52 .61 .45 .32],'FontSize',
,'BackgroundColor',[255/255 153/255 255/255]);

p21 = uipanel(DlgH2,'Title','Features and Blocks','Position',[.52 .32 .45 .22],'
[255/255 153/255 255/255]);

p31 = uipanel(DlgH2,'Title','Stage of Processing','Position',[.52 .1 .45 .18],'E
[255/255 153/255 255/255]);

lb11 = uicontrol('Style','listbox',...
                'String',{'Gama Transform by Gama=5) ','Butterworth 1st order HE
                'Max',2,'Min',0,'Value',[1 3],...
                'Position',[750 600 400 60],'FontSize',11.5);

lb21 = uicontrol('Style','listbox',...
                'String',{'close With disk of radius 6 ','TopHat & BottomHat wit
                'Max',2,'Min',0,'Value',[1 3],...
                'Position',[750 500 400 60],'FontSize',11.5);

lb31 = uicontrol('Style','listbox', 'String',...
                {'Devide Image into 100*150 regions ',''},...
                'Position',[750 350 400 25],'FontSize',11.5);

lb41 = uicontrol('Style','listbox', 'String',...
                {'LBP Histogram ','Co-Occurace matrix statistics','Laws features
                'Position',[750 270 400 60],'FontSize',11.5);

hf1 = waitbar(0,'Generating Feature Vector...');

titleHandle1 = get(findobj(hf1,'Type','axes'),'Title');
set(titleHandle1,'FontSize',12,'HorizontalAlignment','right')

cf1 = get(hf1,'Children');
set(cf1,'Parent',2);
set(cf1,'Units','Normalized','Position',[.59 .15 .31 .05]);
close(hf1);

hf2 = waitbar(0,'1','Name','Extracting features',...
            'CreateCancelBtn',...
            'setappdata(gcbf,''canceling'',1)');

titleHandle2 = get(findobj(hf2,'Type','axes'),'Title');
set(titleHandle2,'FontSize',12,'FontWeight','demi')
```

GUI for training of KSOM:

```
DlgH = figure(1);

set(1,'units','normalized','outerposition',[0 0 1 1]),set(1,'ToolBar','none');
set(1,'MenuBar','none');
set(1,'Name',sprintf('Fabric defects detection'),'NumberTitle','off');

obj12 = uicontrol('Style', 'PushButton','String', 'Exit now','Callback', 'delete(gcbf)','Po
'FontSize',11.5,'FontWeight','bold');

p12 = uipanel(DlgH,'Title','Filtering','Position',[.52 .61 .45 .32],'FontSize',11,'FontWeig
,'BackgroundColor',[255/255 153/255 255/255]);

p22 = uipanel(DlgH,'Title','Features and Blocks','Position',[.52 .32 .45 .22],'FontSize',11
[255/255 153/255 255/255]);

p32 = uipanel(DlgH,'Title','Stage of Processing','Position',[.52 .1 .45 .18],'FontSize',11,
[255/255 153/255 255/255]);

lb12 = uicontrol('Style','listbox',...
                'String',{'Size of SOM Lattice ','Lattice Stucture ',''},...
                'Max',2,'Min',0,'Value',[1 3],...
                'Position',[750 600 400 60],'FontSize',11.5);

lb22 = uicontrol('Style','listbox',...
                'String',{'Neighborhood Function-gaussian ','Radius of Coarse Training -[6
                'Max',2,'Min',0,'Value',[1 3],...
                'Position',[750 500 400 60],'FontSize',11.5);

lb32 = uicontrol('Style','listbox', 'String',...
                {'Iterations(coarse)-10000 ','Iterations(fine)-1000'},...
                'Position',[750 300 400 55],'FontSize',11.5);

h = waitbar(0,'Training of Self-Organizing Map');

titleHandle2 = get(findobj(h,'Type','axes'),'Title');
set(titleHandle2,'FontSize',12,'HorizontalAlignment','right');
cc = get(h,'Children');
set(cc,'Parent',1);
set(cc,'Units','Normalized','Position',[.59 .15 .31 .05]);
close(h);
```