Cairo University
Faculty of Enigeering

Utilizing PDEs In Image Denoising

Presented for MTH 2175 project

Presented to:

Dr. Samah El-Tantawy

TEAM MEMBERS (2nd Year Electronics and Electrical Communication Engineers)	
مجدي احمد عباس عبد الحميد الابرق	Section: 3 / I.D: 9210899
مازن احمد عمر مصطفى عمر	Section: 3 / I.D: 9210887
مازن وائل ضياء الدين احمد رأفت	Section: 3 / I.D: 9210892
محمد ابراهيم محمد على	Section: 3 / I.D: 9210906
فاروق هاشم سعيد عبد اللطيف	Section: 3 / I.D: 9210798
عمر رضا ابراهيم البيومي محمد	Section: 3 / I.D: 9213279
فرح احمد فتحي انور	Section: 3 / I.D: 9210809
منه احمد سيد احمد سيد	Section: 4 / I.D: 9211237

Appendix

MATLAB RGB CODE

```
%% Raw image datastore path
imds = imageDatastore("E:\pics for image");
%% Creatng denoising image data set
inmds  =  denoisingImageDatastore(imds, ’patchSize’, 50);
%% Specifying the training options
options  =  trainingOptions(’sgdm’, …
   ’Momentum’ , 0.9, …
   ’InitialLearnRate’, 1e − 3, …
   ’LearnRateDropFactor’ , 0.2000, …
   ’LearnRateDropPeriod’, 5, …
   ’L2Regularization’,       1.0000e − 04, …
   ’GradientThresholdMethod’, ’l2norm’, …
   ’GradientThreshold’, Inf, …
   ’MaxEpochs’, 3, …
   ’MiniBatchSize’, 64, …
   ’Shuffle’ , ’every − epoch’, …
   ’Plots’ , ’training − progress’);
%% Specifying the network architecture
layers = dnCNNLayers(’NetworkDepth’, 5);
%% Training Our network
net = trainNetwork(inmds, layers, options);
%% Illustrating the raw input and the noisy input
pristineRGB  =  imread("lighthouse. png");
pristineRGB  =  im2double(pristineRGB);
pristineRGB  =  imresize(pristineRGB, [512 512]);
noisyRGB  =  imnoise(pristineRGB, "gaussian", 0,0.01);
[noisyR, noisyG, noisyB]  =  imsplit(noisyRGB);
figure
imshowpair(pristineRGB, noisyRGB, ’montage’);
title(’Original Image (left) and Noisy Image (right)’)
%% Splitting the RGB image into three seperate channels then concatenating the results
denoisedR  =  denoiseImage(noisyR, net);
denoisedG  =  denoiseImage(noisyG, net);
denoisedB  =  denoiseImage(noisyB, net);
denoisedRGB  =  cat(3, denoisedR, denoisedG, denoisedB);
%% Illustrating the output results
figure
subplot(1,3,1);
imshow(I);
subplot(1,3,2);
imshow(denoisedRGB);
subplot(1,3,3);
imshow(denoisedI);
noisyPSNR  =  psnr(noisyRGB, pristineRGB);
fprintf("\n The PSNR value of the noisy image is %0.4f. ", noisyPSNR);
denoisedPSNR  =  psnr(denoisedRGB, pristineRGB);
fprintf("\n The PSNR value of the denoised image is %0.4f. ", denoisedPSNR);
noisySSIM  =  ssim(noisyRGB, pristineRGB);
fprintf("\n The SSIM value of the noisy image is %0.4f. ", noisySSIM);
denoisedSSIM  =  ssim(denoisedRGB, pristineRGB);
fprintf("\n The SSIM value of the denoised image is %0.4f. ", denoisedSSIM);
```

MATLAB GRAYSCALE CODE

```
%% Raw image datastore path
imds = imageDatastore("E:\pics for image");

%% Creatng denoising image data set

inmds  =  denoisingImageDatastore(imds, 'patchSize', 50);

%% Specifying the training options

options  =  trainingOptions('sgdm',...
   'Momentum' , 0.9,...
   'InitialLearnRate', 1e − 3,...
   'LearnRateDropFactor' , 0.2000,...
   'LearnRateDropPeriod', 5,...
   'L2Regularization',       1.0000e − 04,...
   'GradientThresholdMethod', 'l2norm',...
   'GradientThreshold', Inf,...
   'MaxEpochs', 3,...
   'MiniBatchSize', 64,...
   'Shuffle' , 'every − epoch',...
   'Plots' , 'training − progress');
%% Specifying the network architecture

layers = dnCNNLayers('NetworkDepth', 5);
%% Training Our network
net = trainNetwork(inmds, layers, options);

%% Illustrating the raw input and the noisy input

I = imread('cameraman. tif');
I =  imresize(I, [512 512]);
noisyI = imnoise(I, 'gaussian', 0,0.01);
figure
imshowpair(I, noisyI, 'montage');
title('Original Image (left) and Noisy Image (right)')

%% Denoising the input
denoisedI = denoiseImage(noisyI, net);
%% Illustrating the output results
figure
subplot(1,3,1);
imshow(I);
subplot(1,3,2);
imshow(noisyI);
subplot(1,3,3);
imshow(denoisedI);
noisyPSNR  =  psnr(noisyI, I);
fprintf("\n The PSNR value of the noisy image is %0.4f. ", noisyPSNR);
denoisedPSNR  =  psnr(denoisedI, I);
fprintf("\n The PSNR value of the denoised image is %0.4f. ", denoisedPSNR);
noisySSIM  =  ssim(noisyI, I);
fprintf("\n The SSIM value of the noisy image is %0.4f. ", noisySSIM);
denoisedSSIM  =  ssim(denoisedI, I);
fprintf("\n The SSIM value of the denoised image is %0.4f. ", denoisedSSIM);
```

RGB Training DATA SET

GRAYSCALE TRAINING DATA SET