Classification-based Adaptive Image Denoising

CLASSIFICATION-BASED ADAPTIVE IMAGE DENOISING

BY

LAURA McCRACKIN, B.Eng.

A THESIS

SUBMITTED TO THE DEPARTMENT OF ELECTRICAL & COMPUTER ENGINEERING

AND THE SCHOOL OF GRADUATE STUDIES

OF MCMASTER UNIVERSITY

IN PARTIAL FULFILMENT OF THE REQUIREMENTS

FOR THE DEGREE OF

MASTER OF APPLIED SCIENCE

Master of Applied Science (2015) McMaster University
(Electrical & Computer Engineering) Hamilton, Ontario, Canada

TITLE:	Classification-based Adaptive Image Denoising
AUTHOR:	Laura McCrackin B.Eng., (Computer Engineering) McMaster University, Hamilton, Ontario, Canada
SUPERVISOR:	Dr. Shahram Shirani
NUMBER OF PAGES:	xii, 53

For A.V.S., with love.

Abstract

We propose a method of adaptive image denoising using a support vector machine (SVM) classifier to select between multiple well-performing contemporary denoising algorithms for each pixel of a noisy image. We begin by proposing a simple method for realistically generating noisy images, and also describe a number of novel and pre-existing features based on seam energy, local colour, and saliency which are used as classifier inputs. Our SVM strategic image denoising (SVMSID) results demonstrate better image quality than either candidate denoising algorithm for images of moderate noise level, as measured using the perceptually-based quaternion structural similarity image metric (QSSIM). We also demonstrate a modified training point selection method to improve robustness across many noise levels, and propose various extensions to SVMSID for further exploration.

Acknowledgements

First and foremost, I would like to express my gratitude to my supervisor, Dr. Shahram Shirani, with whose guidance and support my vague idea for a fourth-year research project has evolved into SVMSID.

Thank you as well to Seth Ansell, my co-op supervisor at BlackBerry Ltd., who was the first to hear my early thoughts about a denoising algorithm, and encouraged me to seriously explore them.

I would also like to thank my family, lab mates, and friends for their patience, support, and unwavering encouragement.

Notation and abbreviations

Acronyms

AWGN Additive white Gaussian noise.

GTM Ground Truth Map, which indicates the best denoising candidate for each pixel, and is the “correct answer” used to train the classifier.

PSNR Peak signal-to-noise ratio, a mathematically simple (but not perceptually based) means of measuring image quality.

QSSIM Quaternion structural similarity image metric, an extension of the perceptually-based SSIM image quality metric for colour images.

SSIM Structural similarity image metric, a perceptually-based image quality metric designed for single-channel (greyscale) images.

SVMSID Support vector machine strategic image denoising, our proposed denoising method.

SVM Support vector machine.

Variables and Parameters

w An arbitrary window size; when not otherwise specified, $w = 11$.

σ Standard deviation.

$\mathbf{A}_n$ The nth iteration of the denoised image formed by selecting from the denoised image candidates $\mathbf{D}$ according to the ground truth map, $\mathbf{G}_n$.

$\mathbf{G}_n$ The nth iteration of the ground truth map.

γ The free parameter for the RBF classification kernel.

C The misclassification cost parameter of the SVM classifier.

$\mathbf{A}$ The noisy input image.

m The number of denoising candidates used with the classifier, and therefore the number of denoised image versions for each noisy test image.

$\mathbf{A}_{ref}$ The original, noiseless reference image.

$\mathbf{D}$ The set of m denoised image versions.

$\mathbf{M}_{n,k}$ The SSIM map used to calculate the GTM for iteration n.

Contents

Abstract **iv**

Acknowledgements **v**

1 Introduction **1**

2 Algorithm Overview **4**

3 Test Image Generation **8**

3.1 Adding Realistic Image Noise 9

3.2 Quantifying Noise Levels 10

3.3 Test Image Noise Range 11

4 Ground Truth Map Generation **14**

4.1 Error Metric Selection 15

4.2 Denoising Candidate Algorithms and Parameters 16

4.3 First Iteration 16

4.4 Subsequent Iterations 18

4.5 Iterations Required 19

4.6 Parallelised Implementation 21

5 Candidate Features 22

5.1 Novel Features 23

5.1.1 Global seam energy 23

5.1.2 Local seam energy 26

5.1.3 Simple local colour variance 26

5.1.4 Improved local colour variance 29

5.2 Additional features 30

5.2.1 Image signature saliency 30

5.2.2 Sobel filter 31

5.3 Feature selection 32

6 Training, Testing, and Result Evaluation 33

6.1 Training point selection 33

6.2 SVM Kernel Function and Parameters 34

6.3 Result Evaluation 35

7 A Basic Two-Choice Classifier 38

7.1 Results 39

7.2 Computational Complexity 41

7.2.1 Feature Calculations 42

7.2.2 SVM Classification 42

7.2.3 Candidate Denoising Methods 42

7.2.4 Comparison with Traditional Methods 43

8 A Two-Choice Classifier with Multiple Noise Levels 44

8.1 Training Using A Single Noise Level 45

8.2 Training Using Multiple Noise Levels 46

9 Proposed Extensions 48

9.1 Classification Between Multiple Parameters 48

9.2 Hierarchical Classifier 49

10 Conclusions and Further Work 50

List of Figures

2.1 The overall training and testing procedure. 5

3.2 The process of adding realistic noise to a colour image. 9

3.3 Some sample middle grey images with added noise. 11

3.4 The relationship between the simulated noise of a middle-grey image and the resulting ISO 15739 VisualNoise output levels. 12

3.5 Two images with different levels of noise. 13

4.1 Two images with denoising candidates, Ground Truth Maps, and the resulting “ideal” denoised output images. 17

4.2 The effect of each successive Ground Truth Map refinement iteration on the SSIM of the output image. 20

5.1 An image and its global seam energies. 25

5.2 An image and its local seam energies. 27

5.3 An image and its local colour variances. 28

5.4 An image and its Image Signature Saliency and normalized Sobel values. 31

6.1 The results of running a grid search to determine optimal parameter values for a sample experiment, as outputted by LIBSVM’s Grid script. 36

7.1 Denoising results for two test images. 40

8.2 The resulting classifier accuracy and QSSIM when using a classifier trained using only images with a single, moderate level of noise. . . . 47

8.3 The resulting classifier accuracy and QSSIM when using a classifier trained using images with a variety of noise levels. 47

Chapter 1

Introduction

Image denoising is perhaps one of the most mature areas of image processing research, and over the past several decades tremendous strides have been made. Numerous algorithms exist that can provide remarkable recovery from noise on a colour image, with some notable recent examples being the Block-Matching and 3D Filtering algorithm (BM3D) by Dabov et al. [1] and the Non-local Means algorithm (NLM) by Buades et al. [2].

However, even the best algorithms are seldom the best choice for every image, or even every part of an image. An algorithm that can effectively preserve the details in high-frequency areas of an image is likely to leave behind artifacts on more uniform image regions that we would prefer to be smoothed away, while an algorithm that is good at smoothing out uniform areas is unlikely to preserve details that we would rather be largely left alone. In real-world cases, images are seldom homogeneous: consider the sharply-focused details of a subject's face set against a blurred background, or a cloudless blue sky above a lush garden. As humans, we are intuitively aware of this distinction. In image processing, however, this is difficult to account for.

To address this issue, we have created a denoising algorithm that is unlike any other, while building on the strong performance of the best existing methods. We extract a variety of features from each area of an image, which attempt to quantify the colour variance, amount of detail, and other relevant qualities, and use them to train a Support Vector Machine (SVM) classifier to intelligently select which type and strength of denoising algorithm should be used in each image area for the best results.

This algorithm, Support Vector Machine Strategic Image Denoising (SVMSID), was first presented at the 2014 IEEE International Conference on Image Processing in our paper, "Strategic Image Denoising using a Support Vector Machine with Seam Energy and Saliency Features" [3]. In this paper, a basic 2-choice classifier is described, selecting between moderate-strength NLM and BM3D denoising methods for each pixel of a library of test images, all of which contain the same, moderate level of realistically simulated image noise.

It is mainly this basic architecture that we will use to illustrate our denoising method in the sections that follow. We begin with a high-level overview of this method in Chapter 2. The sections that follow describe each portion in further detail. We then provide some results from this denoising method when it is used with a simple two-choice classifier.

In the final chapters, we cover various extensions of this method, which are to be included in an upcoming journal publication. Firstly, we discuss the extension of the simple two-choice classifier to cover images of not only one noise level, but a wide variety of noise levels. We then briefly discuss two extensions that are currently in progress: the use of our method to select between multiple parameters for the same algorithm, as well as the use of a hierarchical classifier to combine the selection between multiple parameters with selecting between multiple denoising algorithms. We then conclude, and briefly discuss some areas for further exploration and improvement.

Chapter 2

Algorithm Overview

We will begin by briefly explaining the high-level structure of our algorithm, and our classifier training and testing procedures. A graphical representation of our method may be seen in Figure 2.1.

Image Generation

To generate a test image set for denoising, realistic image noise is added to each of a set of high-quality, noiseless images.

Training and Testing Image Separation

Images are then divided into two groups – training images, whose Ground Truth Maps will be used to train the classifier, and test images, for which the classifier does not know the correct answer, and which are used to evaluate its performance.

To ensure that testing is not ‘unfair’, it should be noted that noisy images generated from the same base image will always be considered in the same image category;

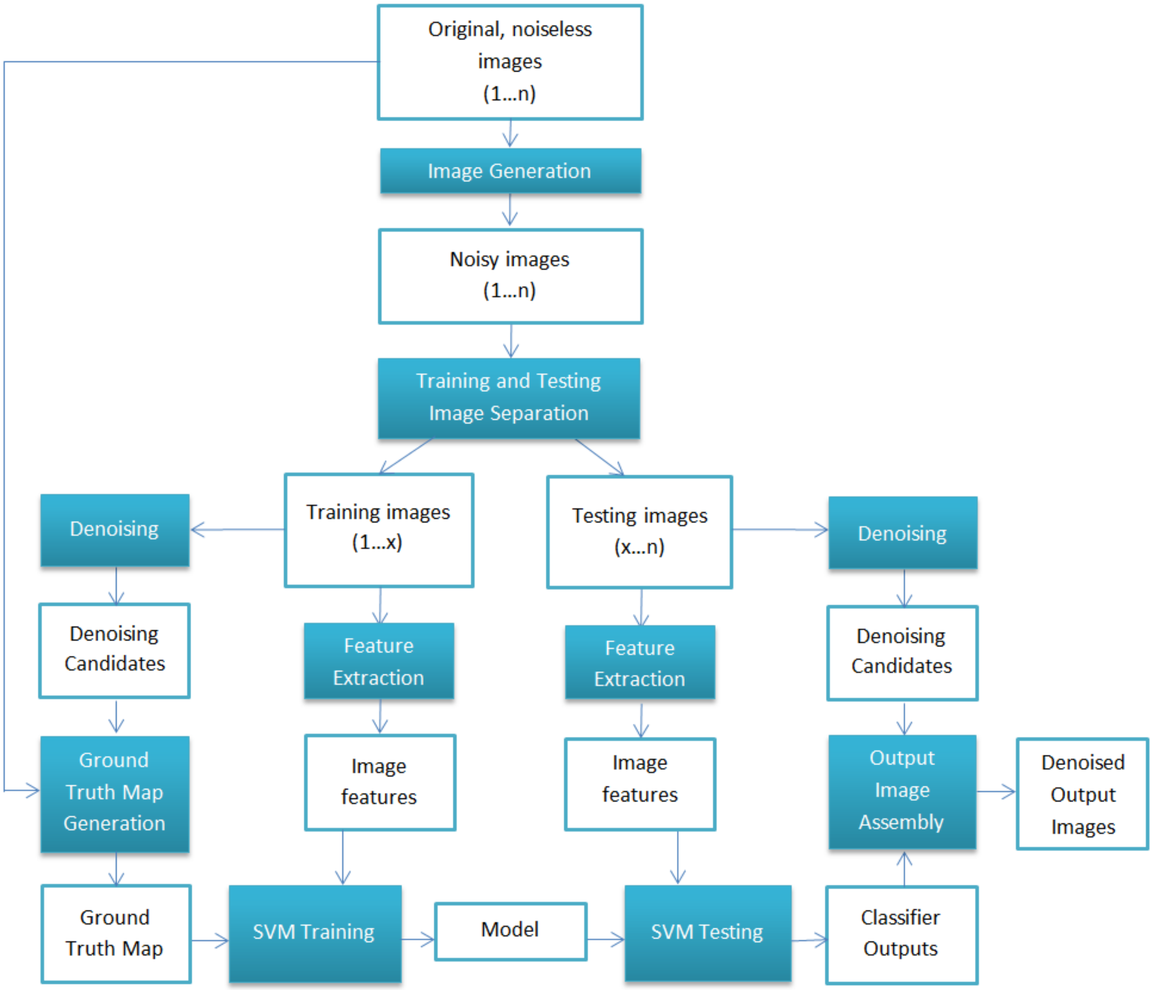


Figure 2.1: The overall training and testing procedure.

that is, as either training images or testing images. The classifier's performance will never be tested using an image that has been included in the training set.

Denoising with Denoising Candidates

Next, each of the two or more candidate denoising algorithms is used to denoise each of the noisy images.

Ground Truth Map Generation

For each pixel location of an image, we determine which of the candidate denoising algorithms gives us the closest value to the original image. This is the Ground Truth Map, which serves as the "correct answer" when classifying.

Feature Extraction

A number of features are extracted from each pixel of the noisy images, which convey information about colour, texture, edge characteristics, and more.

SVM Training

The Support Vector Machine is then trained using a carefully selected group of training points, which are pixels from the training images. The correct answer for these pixels is provided, as taken from the Ground Truth Map, along with the extracted features that are determined to be the most relevant.

With this information, the SVM generates a model that can be used to predict, given values for each feature, what the correct denoising method would be for a new pixel it has not seen before.

SVM Testing

The SVM is then used to predict the correct denoising method for each pixel of the noisy test images based on the extracted image features.

Output Image Assembly

Output denoised images are then formed according to the classifier's output by selecting the appropriate pixels from each denoising candidate. The image quality of these denoised output images may then be evaluated by comparing them to their original, noiseless versions.

With this general overview in mind, we can now move on to discussing each portion of our method in greater detail in the sections that follow.

Chapter 3

Test Image Generation

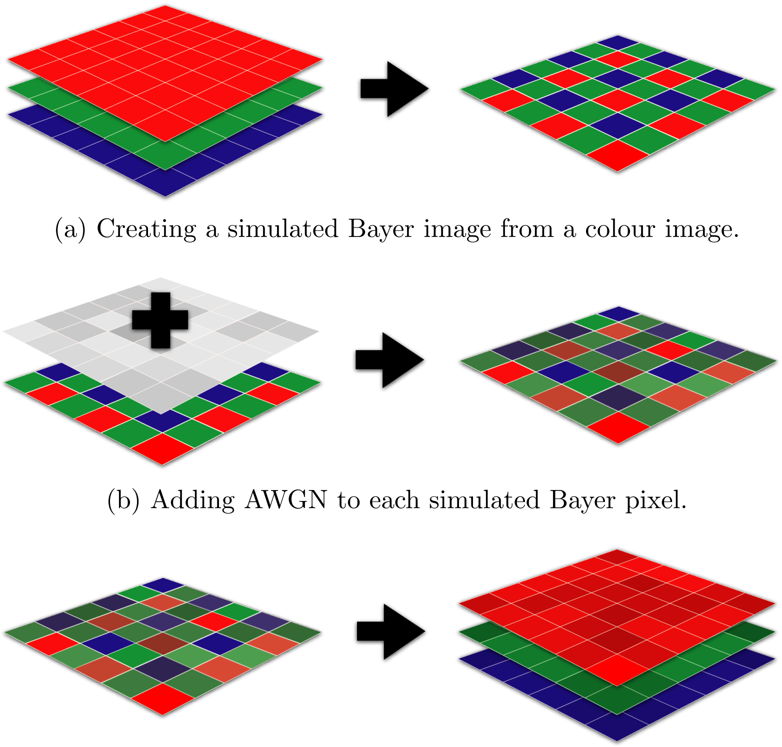
To develop our algorithm, we first create a set of test images. Our test image library is based on the McMaster dataset created by Zhang et al. for their demosaicing research [4].

While it is popular in denoising research to assume that image noise may be modelled as Additive White Gaussian Noise (AWGN) applied to a processed image, this simplistic assumption yields a poor approximation of the noise typically encountered in images [5]. The most commonly observed noise in digital photographs is photon shot noise, which is caused by the random variation in the number of incident photons detected by an image sensor. This variation is particularly noticeable when the total number of incident photons is small, either due to low light conditions or a short exposure time. Low-end image sensors, with smaller pixel sizes, are also particularly affected [3].

3.1 Adding Realistic Image Noise

To realistically simulate the grainy noise typically found in a low-light cell phone photo, it is important to apply noise to unprocessed images. This process is shown in Figure 3.2, and was initially described in our first paper [3].

We begin by forming simulated Bayer images from the McMaster dataset by removing all but one of the colour channels from each pixel of the image. Incident photon noise is then added to the Bayer image, which may be reasonably approximated by using AWGN applied to each Bayer pixel [5]. Finally, the image is demosaiced using gradient-corrected bilinear interpolation to once again form an RGB image [6].



(a) Creating a simulated Bayer image from a colour image.

(b) Adding AWGN to each simulated Bayer pixel.

(c) Demosaicing to form a noisy colour image.

Figure 3.2: The process of adding realistic noise to a colour image.

In this manner, the effect of the photon noise applied to each pixel is spread to neighbouring pixels, as in a real digital photograph. Other sources of noise and noise spread, including pixel nonuniformity, lens shading, colour correction and gamma correction, are comparatively small and are here assumed to be negligible [5].

3.2 Quantifying Noise Levels

A method of accurately simulating noise is more meaningful if there is also a useful way of quantifying the level of noise that is being generated. Certainly, this can be done by specifying the parameters of noise added to the image before the demosaicing process, but these values are not easily comparable to other methods of noise generation, or, perhaps more importantly, the noise actually found in a photograph, which we are attempting to simulate.

The problem of measuring image noise has already been addressed by the Internation Standards Organization (ISO), whose ISO 15739 standard proposed a Visual-Noise scale which is commonly used by the designers and testers of camera systems to compare noise performance [7].

To determine the relationship between our simulated noise and VisualNoise levels, noise was added to a middle grey (with brightness 128 of 255) uniform image 200×200 pixels in size. The main parameter used to control the strength of our simulated noise is the standard deviation, σ, of the Gaussian noise added once the image has been converted to a simulated Bayer image. Resulting noisy images were then processed using official ISO 15739 software using a 200×200 region of interest and default viewing conditions to obtain their VisualNoise levels. Sample middle grey images at selected VisualNoise levels are shown in Figure 3.3. This method was designed

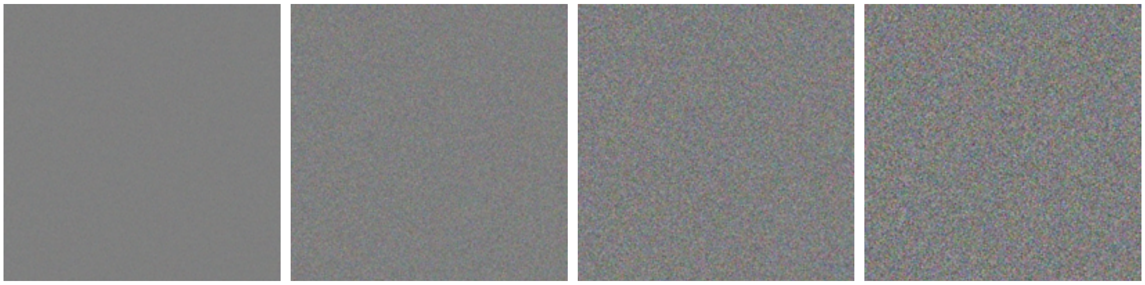


Figure 3.3: Some sample middle grey images with added noise. *From left to right:* VisualNoise = 2; VisualNoise = 7; VisualNoise = 12; VisualNoise = 17.

to simulate what is done in the industry to test the noise performance of camera systems: uniform grey regions of a test chart are photographed in controlled lighting conditions, and the noise present in the resulting photographs is then calculated.

The associated VisualNoise levels for σ values between 0 and 0.005 may be seen in Figure 3.4. It is interesting to note that the relationship is nonlinear for lower values of σ – the region we are most interested in – and appears to become linear for higher values. Images with higher σ values can also be seen to deviate further from the curve, which intuitively makes sense since there is a larger range of potential output values for pixels with stronger noise.

It is using this graph as a lookup table that we are able to determine appropriate simulated noise values for our test images.

3.3 Test Image Noise Range

In our first paper [3], only one noise level was used for all training and test images. A value of $\sigma = 0.0008$ was used for the standard deviation of the AWGN added to our simulated Bayer images; this correlates with an ISO 15739 VisualNoise value of approximately 8 and was judged to be a moderate level of noise, characteristic of noticeable but not extreme image degradation.

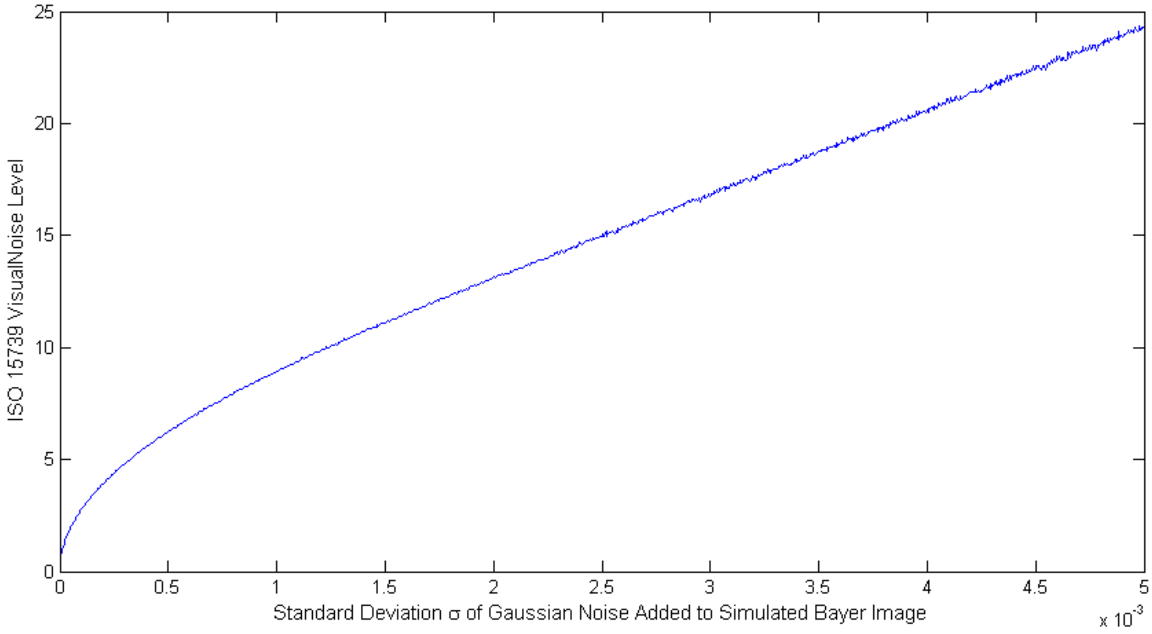


Figure 3.4: The relationship between the simulated noise of a middle-grey image (specifically, the standard deviation of the AWGN added to the simulated Bayer form of the image) and the resulting ISO 15739 VisualNoise output levels.

However, the amount of noise present in a typical mobile camera photo can vary greatly. To better highlight the robustness of our approach, our test image collection has since been expanded to include versions of each image at several different levels of noise, ranging from no perceptible noise to extreme image degradation. The σ values for these images were selected to correspond with ISO 15739 VisualNoise levels of 2, 5, 8, 11 and 14, respectively.

Two images from the test image library, in their original state and with noise added, may be seen in Figure 3.5.

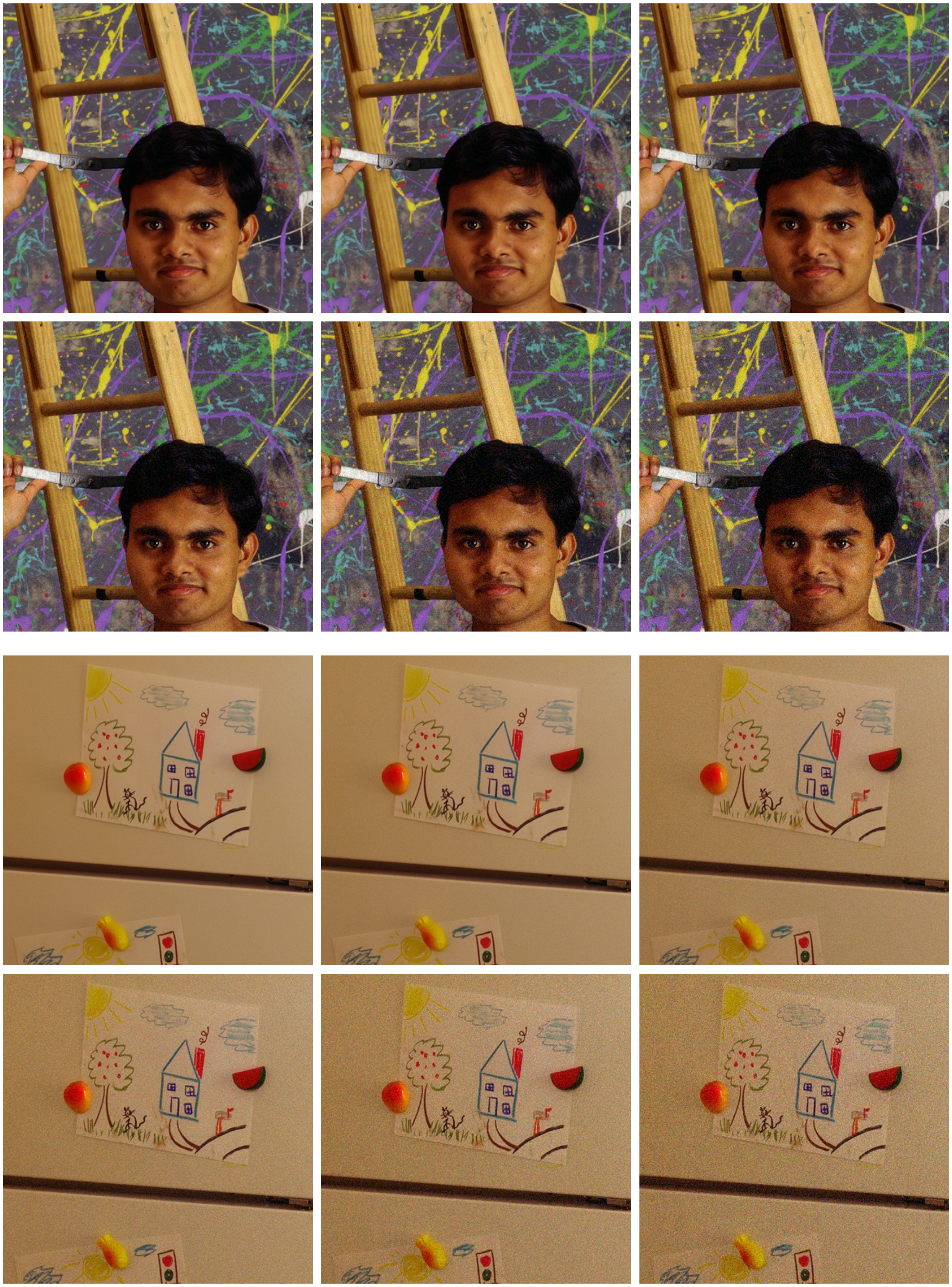


Figure 3.5: Two images, with different levels of noise. *From top left to bottom right:* original, noiseless image; VisualNoise = 2; VisualNoise = 5; VisualNoise = 8; VisualNoise = 11; VisualNoise = 14.

Chapter 4

Ground Truth Map Generation

Generating the Ground Truth Map (GTM) for denoising images – that is, deciding which denoising candidate algorithm and strength is most appropriate for every pixel of an image – is an important consideration, as these are the maps that will be provided to the support vector machine classifier for each noisy image as being the "correct answer" for training purposes. At the most basic level, this means determining which algorithm and parameter combination results in the lowest error for each pixel, which is quantitatively measured as the highest possible similarity between the denoised pixel region and that of the original, noiseless image.

The SSIM algorithm by Wang et al. provides us with an output map of the denoised image's perceptual similarity to the original image, as a matrix of values between 0 (entirely dissimilar) and 1.0 (identical) [8]. Because the SSIM is measured on a block-by-block basis using an 11×11 pixel region around the pixel of interest, selecting the best denoising method for a pixel cannot be done independently of a pixel's neighbours. Each pixel of the image must be assigned denoising parameters, and by selecting denoising method parameters for the neighbours of a pixel of interest, we affect not only the SSIM of the pixel of interest, but we may also change the ideal denoising parameters for that pixel.

Because of this interdependence, generating the ground truth is not a simple task involving a single calculation step for each pixel. In fact, generating an ideal GTM is an intractable problem. For our purposes, however, we make do with a reasonable guess that still allows us to obtain very good denoising performance.

Two sample images are shown in Figure 4.1, each with two denoising candidates, their Ground Truth Map, and the "ideal" denoised image produced by using this Map to select between the denoising candidates for each pixel.

4.1 Error Metric Selection

Since we are working with colour photographs, it should be noted that our first inclination was to measure output image quality using the recently proposed quaternion structural similarity measure (QSSIM) by Kolaman and Yadid-Pecht, an extension of the single-channel SSIM measure for colour images which has been shown to provide a more accurate measure of image quality [9]. However, QSSIM is a much more computationally intensive algorithm than SSIM, and while this additional cost is generally acceptable, the GTM generation in SVMSID requires that this metric be calculated many times per pixel and per iteration. Preliminary testing showed little difference between the quality of output image produced using QSSIM and using SSIM, and so in the interest of minimizing computational complexity, SSIM was chosen as the ground truth map error metric.

It should be noted that SVMSID could also be used with non-perceptual images – for instance, MRI images – in which case a metric like PSNR might be preferred to

SSIM or QSSIM.

4.2 Denoising Candidate Algorithms and Parameters

While any denoising methods may be used as denoising candidate classes for our algorithm, we selected two popular, contemporary choices in the field:

1. Block-matching and 3D filtering method (BM3D), by Dabov et al. [1]
2. Non-local means (NLM), by Buades et al. [2]

A detailed discussion of these algorithms is omitted here for brevity. The strength parameter value for each was selected to be the default value recommended by the literature.

4.3 First Iteration

As a first step, denoising is performed on each training image with each candidate denoising algorithm for all candidate parameter values. A range of m reasonable values is selected per algorithm, which results in a set of m denoised images, $\mathbf{D} = \{\mathbf{D}_1, \mathbf{D}_2, ..., \mathbf{D}_m\}$. For each of these denoised images, a first iteration SSIM map, $\mathbf{M}_1$, of the error at each pixel location when compared to the original, noiseless reference image $\mathbf{A}_{ref}$ is calculated:

$$\mathbf{M}_{1,k}(x, y) = SSIM(\mathbf{D}_k(x, y), \mathbf{A}_{ref}(x, y)), \qquad k = 1, 2, ..., m \quad (4.1)$$

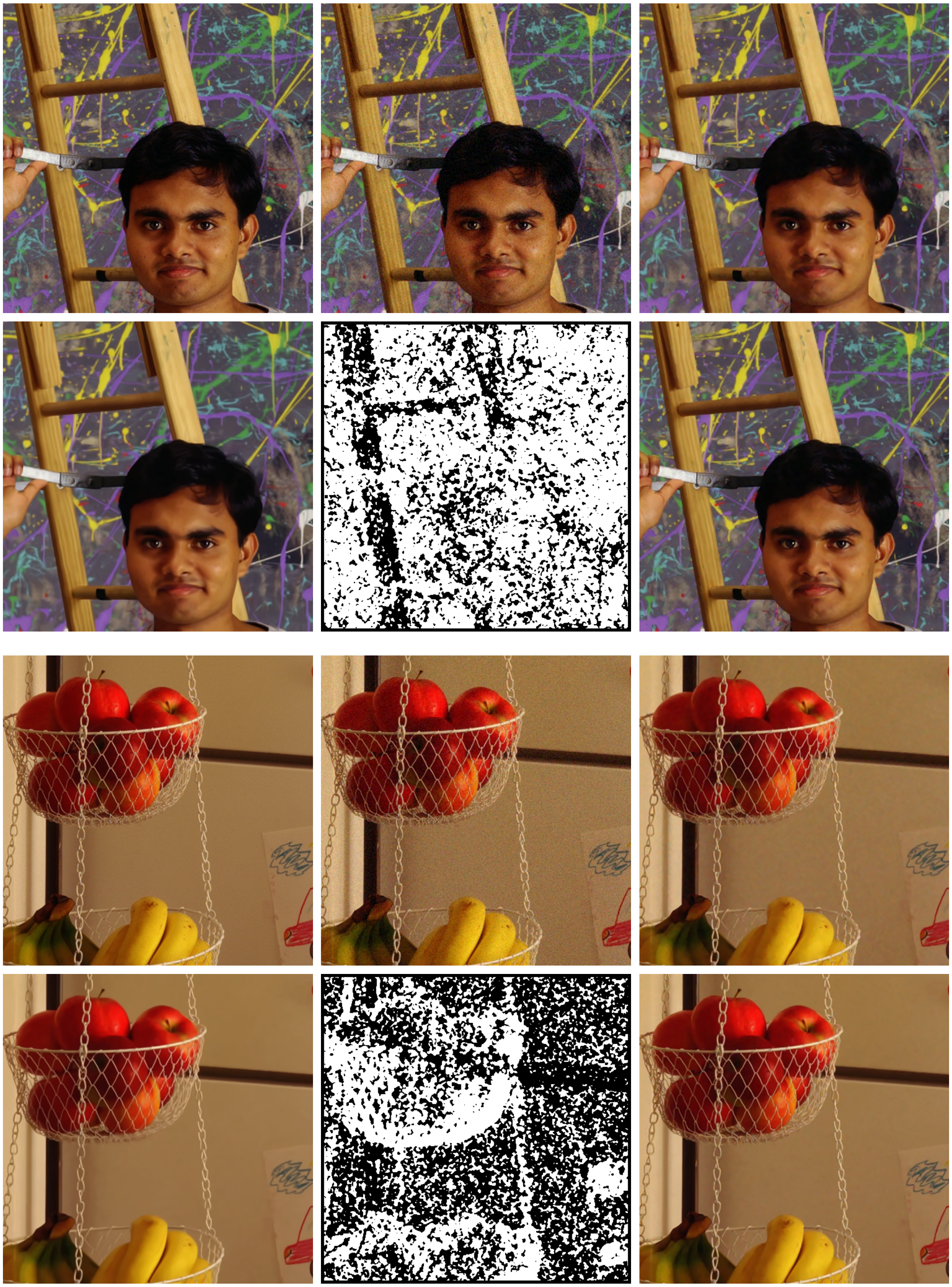


Figure 4.1: Two images with denoising candidates, Ground Truth Maps, and "ideal" denoised output images. *From top-left to bottom-right:* original image, noisy image, NLM denoising, BM3D denoising, GTM (iteration 1; white = NLM, black = BM3D), and the "ideal" denoised image.

Then, a first pass of the ground truth map, $\mathbf{G}_1$, is generated. Pixels are considered independently, one by one, and the candidate denoising parameter that provides the highest SSIM value – that is, the lowest error – for each location is selected:

$$\mathbf{G}_1(x,y) = \underset{k=1,2,...,m}{\arg\max}(\mathbf{M}_{1,k}(x,y)) \tag{4.2}$$

The first iteration of the denoised image, $\mathbf{A}_1$, is then formed by selecting from the denoised image candidates according to the ground truth map for each pixel location, as follows:

$$\mathbf{A}_1(x,y) = \mathbf{D}_{\mathbf{G}_1(x,y)}(x,y) \tag{4.3}$$

4.4 Subsequent Iterations

As mentioned previously, while this first iteration gives us a decent denoised image output, the result can be considerably improved by further iteration. New SSIM maps, $\mathbf{M}_{n,k}$ (where n is the current iteration number, for any $n > 1$), are calculated by substituting alternative denoising candidate choices for the current pixel of interest into the denoised image found in the previous iteration, as follows:

$$\mathbf{M}_{n,k}(x,y) = SSIM(\mathbf{B}_{k,x,y}(x,y), \mathbf{A}_{ref}(x,y)), \qquad k = 1,2,...,m \tag{4.4}$$

where

$$\mathbf{B}_{k,x,y}(i,j) = \begin{cases} \mathbf{D}_k(i,j), & i = x, j = y \\ \mathbf{A}_{n-1}(i,j), & otherwise \end{cases} \tag{4.5}$$

Then, the current iteration's GTM is calculated as before, except that the old pixel value is kept if it does not make a significant difference to the output quality. Specifically, the GTM is updated to the current best denoising choice only if it provides a SSIM improvement over the previous choice that is greater than a threshold T:

$$\mathbf{G}_t(x,y) = \underset{k=1,2,\ldots,m}{\arg\max}(\mathbf{M}_{n,k}(x,y)) \tag{4.6}$$

$$\mathbf{G}_n(x,y) = \begin{cases} \mathbf{G}_t(x,y), & \mathbf{M}_{n,\mathbf{G}_t(x,y)}(x,y) - \mathbf{M}_{n-1,\mathbf{G}_{n-1}(x,y)}(x,y) > T \\ \mathbf{G}_{n-1}(x,y), & otherwise \end{cases} \tag{4.7}$$

This thresholding is performed to prevent the ground truth map from oscillating back and forth between very similar choices. As with the first iteration, the ideal denoised image from iteration n is equal to

$$\mathbf{A}_1(x,y) = \mathbf{D}_{\mathbf{G}_1(x,y)}(x,y). \tag{4.8}$$

This process may be repeated until the desired number of iterations have been performed.

4.5 Iterations Required

Theoretically, the greater the number of iterations performed, the higher the output quality of the resulting GTM should be. In practice, however, the benefit of performing subsequent iterations quickly diminishes as the output converges. To determine an appropriate tradeoff between the number of iterations run and the output performance, GTMs for all test images between ten different parameter values of the NLM

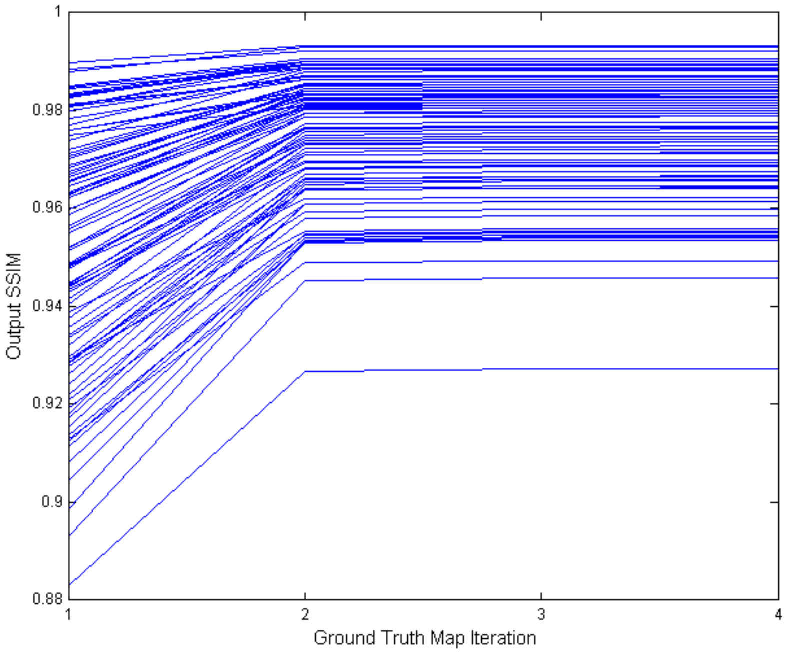


Figure 4.2: The effect of each successive GTM refinement iteration on the SSIM of the output image produced from that iteration's Map. Each curve represents a separate test image.

denoising algorithm were generated. The accuracy for the first four iterations may be seen in Figure 4.2.

Most test images saw a substantial SSIM improvement during the first 2 iterations. The difference between iterations 2 and 3, however, was much smaller, and most images showed no improvement at all between iterations 3 and 4.

An even greater consideration, however, is the effect of the iteration on classifier training and testing accuracy. Preliminary testing showed that using an iteration 3 or 4 GTM did not increase accuracy, but in fact caused a significant decrease in accuracy. This is not entirely surprising when we consider that, with each successive iteration, the class of a pixel becomes more dependant on the classes of its neighbours (some of which may be relatively distant) and less strongly correlated with the image features we are using to train and test the classifier.

Therefore, it was decided that the Ground Truth Maps should be calculated to 2 iterations – that is, a first iteration and a single iteration of refinement.

4.6 Parallelised Implementation

It should be noted that performing SSIM calculations for m possible denoising candidates for n iterations for every pixel of an image quickly becomes a rather computationally intensive undertaking.

Because we are interested in the pixel-by-pixel SSIM for ground truth map generation, rather than the mean SSIM for the entire image, and because SSIM is a block-based method, we can easily improve the throughput of the official SSIM implementation by processing composites of multiple pixels of interest at once. Since the window size used for SSIM calculations is 11 pixels, each composite image contains pixels with x coordinates α, $11 + \alpha$, $22 + \alpha$, and so forth, and similarly with y coordinates α, $11 + \alpha$, $22 + \alpha$, and so on.

This way, rather than naively finding the SSIM of every pixel of interest individually, an entire image's values can be calculated using only 11×11 calls to the SSIM software, for $\alpha = 1, 2, ..., 11$.

Chapter 5

Candidate Features

We now briefly discuss a number of algorithms for extracting potential features from the test images. The first three – global seam energy, local seam energy, and local colour variance – were initially proposed in our first paper [3]. Additional features from the literature are also briefly described. We also discuss our methods for selecting the most appropriate features from these for the SVM.

Many of these features use an arbitrary window size, $w \times w$, around the pixel of interest. When not otherwise specified, a value of $w = 11$ is generally used to coincide with the window size used by the SSIM error metric.

Several of these candidate features have a direction associated with them, and can therefore be calculated on an image in either a horizontal or vertical direction. Both of these variations were considered as separate features for training and testing.

Additionally, to provide further localized information about a pixel's nearest neighbours, the outputs of most features can also be Gaussian blurred, effectively averaging the feature output for the pixel of interest with the output for its neighbours. Gaussian-blurred versions of many features were also considered separately.

A Gaussian window size of 11 × 11 pixels was chosen, as mentioned previously, to coincide with the SSIM window size.

5.1 Novel Features

5.1.1 Global seam energy

First, we define a method based on the seam carving algorithm by Avidan and Shamir for content-aware reduction of image size [10] [3].

The image is first converted to a greyscale image by converting it to the L*a*b* colour space and discarding the a* and b* channels. Each pixel of the image must then be assigned an energy value. While many metrics may be used to assign pixel energy, Sobel edge detection was selected here, as this is both a popular metric and one that is relatively simple to calculate.

To obtain energy values for the image, the image was convolved with a Sobel filter in either the horizontal or vertical direction [11]:

$$\mathbf{G}_x = \begin{bmatrix} +1 & 0 & -1 \\ +2 & 0 & -2 \\ +1 & 0 & -1 \end{bmatrix} * \mathbf{A} \; \textit{and} \; \mathbf{G}_y = \begin{bmatrix} +1 & +2 & +1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} * \mathbf{A} \tag{5.1}$$

where $\mathbf{G}_x$ corresponds to the magnitude of the image gradient in the horizontal direction, and $\mathbf{G}_y$ in the vertical direction. Finally, the absolute value of each Sobel value was taken, so that each value represents the magnitude of pixel change, independent of whether the gradient is in the positive or negative direction.

Next, optimal paths are calculated through the image. Starting with the second row, and for each successive row, each pixel value $p(x, n)$ is replaced with the sum of the pixel's value and the minimum cumulative sum from the three nearest pixels of the previous row, $p(x-1, n-1)$, $p(x, n-1)$, and $p(x+1, n-1)$.

These cumulative sums represent the total energy of potential seams, and the optimal seam, which has the lowest cumulative energy sum in the last row of the image, will then be selected for removal. Backpropogation is used to trace from the seam's endpoint to the beginning using the image's cumulative energy values.

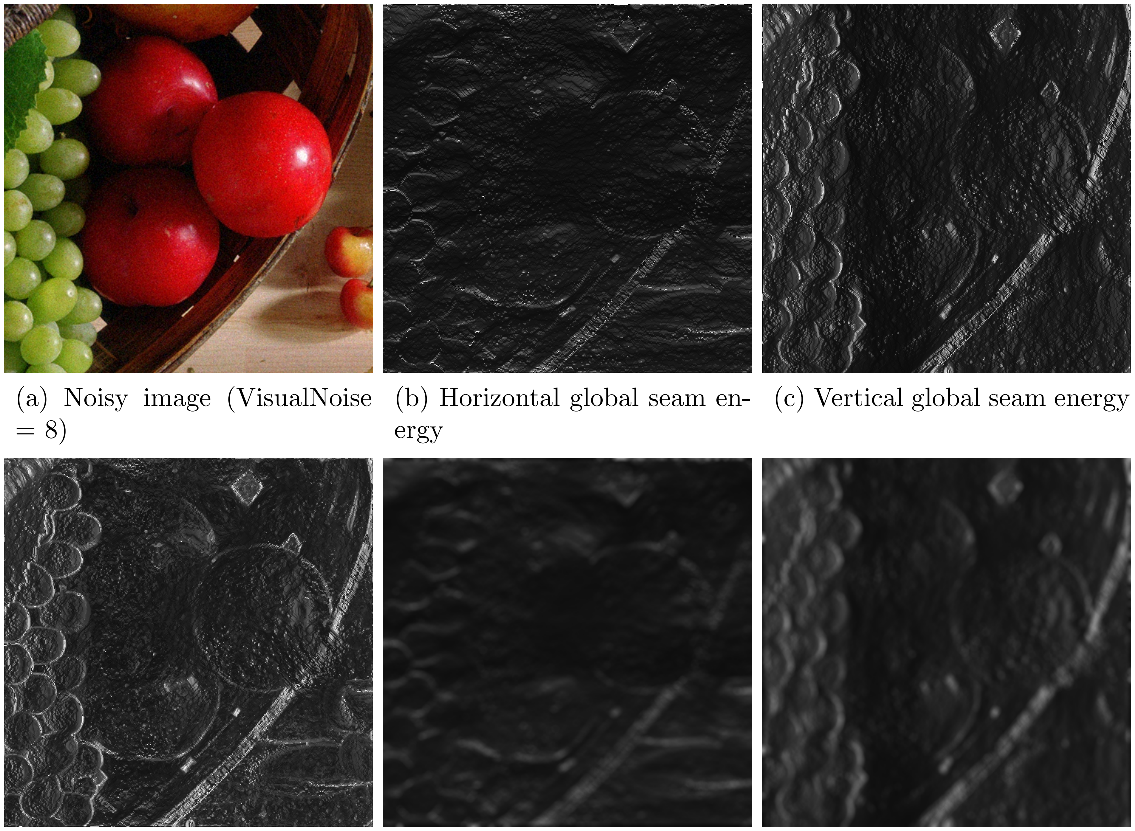
This is performed similarly to the original path calculation. Beginning with the seam coordinate of the last row of the image, and working backwards, the seam coordinate of the previous row is selected to be the lowest cumulative energy value of the 3 pixels closest to the current row's seam coordinate. By backpropogating through to the first row of the image, all coordinates of the desired seam are found.

Finally, the chosen seam is removed from the image. This is done by moving all image pixels to the right of the seam one pixel to the left. The cumulative energy sum of this removed seam is saved as the Global Seam Energy Value for each pixel in the seam.

To remove another seam, these steps are then repeated on the new image, and so on, until the resulting image is of the desired dimensions. For our purposes, we continue until the $m \times n$ image is of size $m \times 2$ (for finding the horizontal seam energy) or $2 \times n$ (for finding the vertical seam energy). By this time, nearly every pixel has been assigned a Global Seam Energy Value. The pixels that remain may be assigned an arbitrary Global Seam Energy Value that is slightly higher than that of the final removed seam.

Several variations of this algorithm as used as potential parameters. Firstly, global seam energy may have either a horizontal or vertical direction, depending on which Sobel filter was used and in which direction seams were removed. To combine both of these values into a single feature, a normalized global seam energy may be defined by simply averaging both the horizontal and vertical global seam energy values. Finally, the horizontal and vertical seam energies may be Gaussian blurred, which allows this feature to convey not only the seam energy of the pixel of interest, but also that of the nearest neighbouring pixels.

Each of these variants may be seen for a sample image in Figure 5.1.



(a) Noisy image (VisualNoise = 8)

(b) Horizontal global seam energy

(c) Vertical global seam energy

(d) Normalized global seam energy

(e) Gaussian-blurred horizontal global seam energy

(f) Gaussian-blurred vertical global seam energy

Figure 5.1: An image and its global seam energies.

5.1.2 Local seam energy

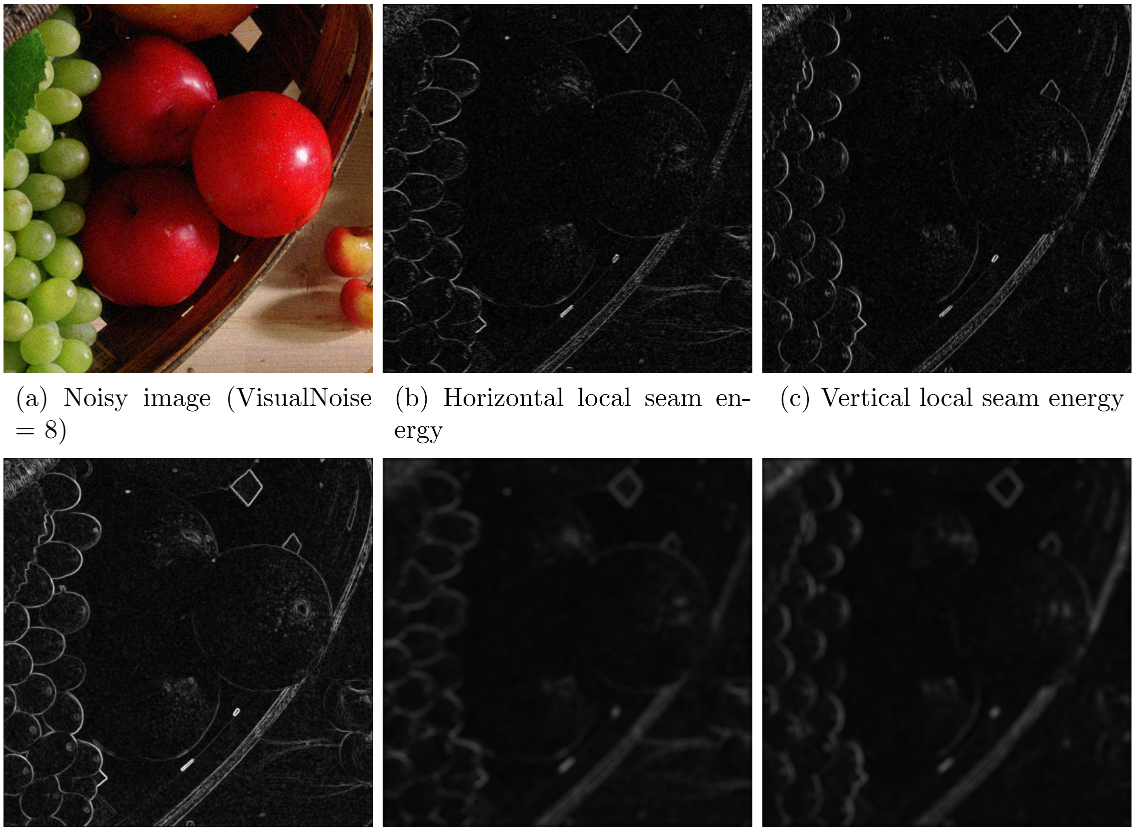
The Local Seam Energy of an image is designed to resemble the Global Seam Energy, but it provides more localized insight into image structure [3].

As before, the image is first converted to a greyscale image, and Sobel edge detection is performed. For each pixel of the test image, a $w \times w$ window of nearest neighbours is then considered. Using the same path calculation algorithm described above, the path of lowest cumulative energy through the centre pixel from the top of the window to the bottom (or from left to right, for horizontal local seam energy) is found. This total cumulative energy is the pixel's local seam energy.

The window is shifted accordingly for the next pixel of interest, and the local seam energy is calculated for all pixels in the image.

5.1.3 Simple local colour variance

Colour variation is an important visual cue suggesting the amount of noise present in an image: picture a photograph of a cloudless blue sky stippled with grainy, coloured artifacts. Furthermore, it is an indication of how much denoising an image region can tolerate; while this cloudless sky might very well benefit from very strong denoising, a highly detailed region with many different colours present in a small area would likely be even further distorted by heavy denoising. To attempt to gauge the amount of colour variation present in an image region, we designed for our original two-choice classifier a feature that measures the local variation in colour differences [3]. We will refer to the algorithm described in this section as the simple local colour variance, as an improved, more perceptually relevant local colour metric is described in the next section.



(a) Noisy image (VisualNoise = 8)

(b) Horizontal local seam energy

(c) Vertical local seam energy

(d) Normalized local seam energy

(e) Gaussian-blurred horizontal local seam energy

(f) Gaussian-blurred vertical local seam energy

Figure 5.2: An image and its local seam energies.

First, the RGB test image is converted to the L*a*b* colour space. Then, for each pixel, the surrounding $w \times w$ window is considered. The CIE76 ΔE^*_{ab} colour difference is then found between each pixel i in the window and the centre pixel of interest c, where [12]

$$\Delta E^*_{ab} = \sqrt{(L^*_i - L^*_c)^2 + (a^*_i - a^*_c)^2 + (b^*_i - b^*_c)^2} \tag{5.2}$$

Any ΔE^*_{ab} values that are less than 2.3 are set to 0, as this is considered to be the point of Just Noticeable Difference (JND), below which a difference in colour is not visually perceptible [12].

$$\mathbf{d}(i,j) = \begin{cases} \Delta E^*_{ab}(i,j), & \Delta E^*_{ab}(i,j) > 2.3 \\ 0, & otherwise \end{cases} \tag{5.3}$$

Next, the variance of all ΔE^*_{ab} values for the $w \times w$ window is calculated. This is the output for the pixel of interest:

$$\mathbf{v}(i,j) = Var(\mathbf{d}(i-w:i+w, j-w:j+w)) \tag{5.4}$$

The window will then be shifted, and the calculation is repeated for the remainder of the pixels in the image. Once the local colour variance has been calculated for all pixels, the values should be normalized.

It should be noted that while the CIE76 ΔE^*_{ab} has been superseded by the more perceptually uniform ΔE^*_{94} and ΔE^*_{00} colour difference formulae, we prefer the older formula here due to its comparative computational efficiency [13].

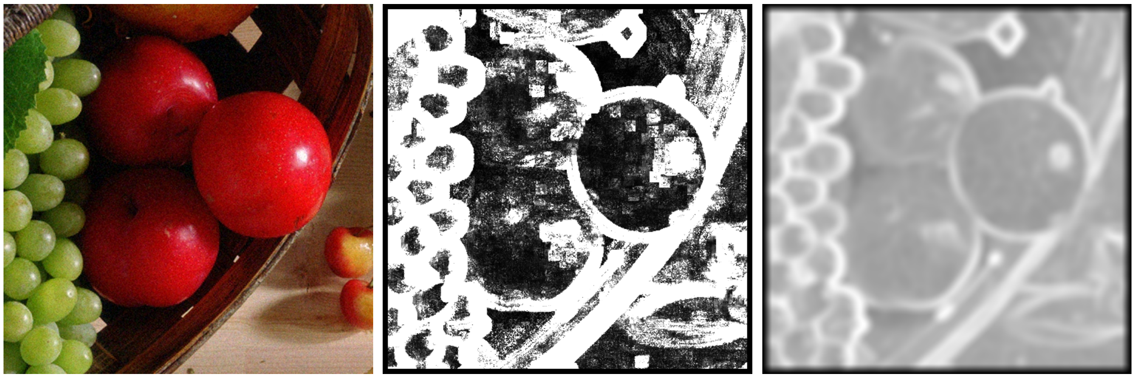


Figure 5.3: An image and its local colour variances. *From left to right:* original, noisy image (VisualNoise = 8); simple local colour variance; improved local colour variance.

5.1.4 Improved local colour variance

While the local colour variance metric originally developed for use with the two-category classifier exhibited good correlation with classification results and little redundancy with other features, it is nonetheless a rather simple metric that fails to take into consideration numerous perceptual phenomena. For that reason, an improved local colour variance metric was developed for use with the hierarchical classifier, and was shown to be a more useful metric than the original colour variance metric according to mRMR feature selection, as described in the last section of this chapter.

Firstly, one limitation of the previous metric was that all pixels at locations (x, y) within the window of interest around the pixel at (i, j) were considered equally, when it is obvious that a pixel located on the edge of the window should not have as much importance as pixels located right next to the pixel at (i, j). A simple weighting method, $\mathbf{w}_{i,j}$, was therefore used:

$$\mathbf{w}_{i,j}(x, y) = \sqrt{2 * \mathbf{d}_{max}{}^2} - \sqrt{|i - x| + |j - y|} \tag{5.5}$$

where $\mathbf{d}_{max}$ is the maximum horizontal or vertical distance from pixel (i, j) within the window, and equals $(0.5 \times w) - 1$. This formula assigns each pixel location with the window a weighting inversely proportional to its distance from (i, j); note that for pixels in the diagonal corners of the window, where both $|i - x|$ and $|j - y|$ are equal to $\mathbf{d}_{max}$, the assigned weight $\mathbf{w}$ will be zero.

Additionally, while a uniform weighting scheme between the variances of all three colour channels – L*, a*, and b* – was originally used, a study by Keelan et al. demonstrates that more perceptually meaningful results can be attained from using an alternative weighting scheme [14]. They experimentally derive an objective metric, $\mathbf{\Omega}$, to reflect the amount of visually perceptible noise in an image. This metric is defined as follows:

$$\mathbf{\Omega} = log_{10}[1 + 100 * \sigma^2(\mathbf{L}^*) + 5 * \sigma^2(\mathbf{a}^*) + 12 * \sigma^2(\mathbf{L}^*\mathbf{a}^*)] \tag{5.6}$$

It is interesting to note that the variance of the b* channel was found to be of negligible importance, while the covariance between the L* and a* channels (but between no other combination of channels) was found to be significant to this metric.

Therefore, the colour variance algorithm was updated to include these findings, and using the weighted the L* and a* values described previously.

5.2 Additional features

In addition to the features described above, we also select some previously defined features from the literature.

5.2.1 Image signature saliency

In psychology and computer vision, saliency is a concept concerning importance; an object that is salient stands out amidst its neighbours and is attributed greater significance. In the context of image processing and computer vision, saliency is an active area of research, particularly in object recognition and similar applications. In our case, saliency is chosen as a feature in the hopes that it provides some insight into the best denoising strategy – perhaps regions that are the most salient are those which we would prefer to largely leave alone, which less salient areas would look more visually pleasing when more heavily denoised.

We use a saliency map calculated by the Image Signature algorithm, developed by Hou et al. as a metric to predict the human fixation points of an image [15]. In this algorithm,

$$\begin{aligned} \mathbf{m} &= g * (\overline{A} \circ \overline{A}) && (5.7) \\ \overline{A} &= IDCT(sign(DCT(A))). && (5.8) \end{aligned}$$

where $\mathbf{m}$ is the image saliency map, A is the input image, and g is a Gaussian kernel.

The Image Signature Saliency for a sample image is shown in Figure 5.4.

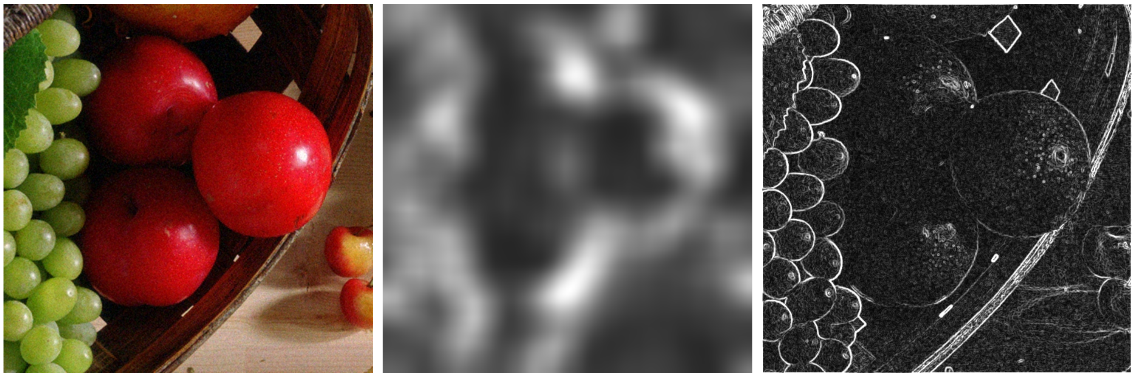


Figure 5.4: *From left to right:* original, noisy image (VisualNoise = 8); Image Signature Saliency; normalized Sobel values.

5.2.2 Sobel filter

Edges also provide us information about the structure of an image, and the amount of denoising that may be tolerated in each area. It is visually preferable to have distinct, clear edges, and so an algorithm that is better at preserving edges would be our best bet for the most prominent edge regions. There are a number of edge detection methods in the literature; we use the Sobel filter, which was also the kernel used to define our seam energy metrics. We use a normalization of the horizontal and vertical Sobel values of the image, as shown in Figure 5.4.

5.3 Feature selection

In the previous sections, we described a number of features to consider for classification. Since SVM training and testing becomes much more computationally and time intensive as additional features are added, we wish to reduce the number of features to only those that will give us the best results.

The algorithm minimum-redundancy maximum-relevance (mRMR) by Peng et al. is used to determine which combination of features best provides the maximum amount of relevant information to the training point class in a small-sized feature set [16].

The top 5 to 6 features according to mRMR are the features selected as inputs for each SVM classifier. The exact features preferred for each classifier will be mentioned in each experiment's corresponding chapter.

Chapter 6

Training, Testing, and Result Evaluation

As mentioned previously, the SVM is used to predict, based on image features, which of multiple possible denoising methods is the most suitable choice for each pixel of the image.

6.1 Training point selection

One of the most important considerations in training support vector machines is the careful selection of training points. The strongest classifier results are usually obtained using a high proportion of difficult or borderline points for training, so that the support vector machine is taught the subtleties of differentiation instead of simply learning the "easy" cases.

In our case, borderline classification cases would be those where the gain in SSIM by selecting one denoising method over the other are minimal. However, these are also the points for which we care least about a correct classification result, since there is little difference in accuracy either way. For our simple two-choice classifier, we therefore select pixels which exhibit an SSIM improvement of between 0.06 and 0.16 using the selected denoising algorithm over the alternative. For each training image, we select 40% of the pixels within this range for training.

6.2 SVM Kernel Function and Parameters

For classification, we use a support vector machine with the popular LIBSVM library [17]. We use a radial basis function (RBF) classifier kernel [18]:

$$K(\mathbf{x}, \mathbf{x}') = exp\left(-\frac{1}{2\sigma^2}||x - x_i||^2\right) \tag{6.1}$$

where a free parameter, γ, is defined as follows:

$$\gamma = \frac{1}{2\sigma^2}. \tag{6.2}$$

A particular optimal value of γ should be selected for the kernel to ensure good classification performance. Additionally, SVM classification has an associated parameter, C, which represents the misclassification cost. A lower C value is more permissive of misclassifying points to allow for a simpler model complexity, which a higher C value will allow for much greater model complexity to ensure a better fit to each data point (and, consequently, means that the time required for training and testing is also greater) [19].

Selecting appropriate values for γ and C is often very important for good classifier performance [19]. This is, however, difficult to do without experimenting with many different combinations of values on the data set to be classified.

Therefore, a grid search was performed on a subset of the training data to find the optimal parameter values using the LIBSVM Grid script [17]. Different combinations of C and γ were systematically tried, with each successive parameter value exponentially greater than the previous one. We use 5-fold cross-validation – that is, randomly partitioning the training points into 5 equal sets, using 4 for training and 1 for testing, and then repeating this process and averaging the classification accuracy between sets. The combination of parameters found to provide the highest cross-validation accuracy is then selected for use with the full training and testing sets.

The output from a grid search for a sample experiment (training and testing a two-choice classifier using images of multiple noise levels, as described in Chapter 8) may be seen in Figure 6.1. The highest accuracy found during the grid search was for parameter values $C = 32768$ and $\gamma = 8$. However, training and testing would take an extremely long time using a C value this large; instead, by looking at the graph, we select parameter values $C = 512$ and $\gamma = 2$ – that is, $log_2(C) = 9$ and $log_2(\gamma) = 1$ – which provide very similar performance at a much lower computational cost.

6.3 Result Evaluation

Traditionally, the mark of a good SVM is strong classification accuracy – that is, the classifier output should match with the “correct” answer according to the Ground Truth Map.

In our case, however, this is not actually the main concern; our measure of success

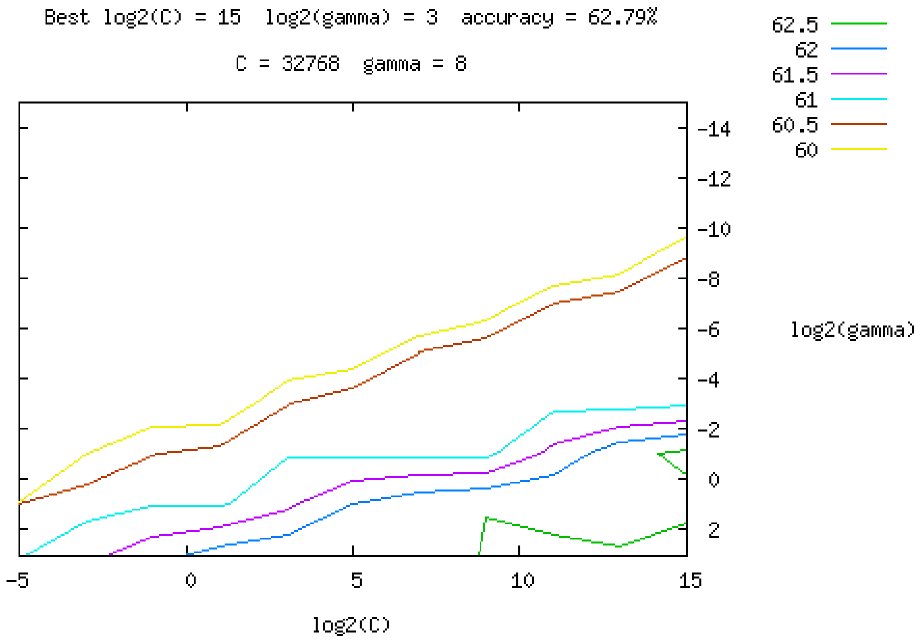


Figure 6.1: The results of running a grid search to determine optimal parameter values for a sample experiment, as outputted by LIBSVM's Grid script. The different colours represent the different levels of cross-validation accuracy for each tested combination of parameters

is that the image produced using the output denoising choice map is visually superior to using either of the denoising methods alone. Since both algorithms perform well, this is possible even if a large number of pixels are classified "incorrectly". Furthermore, because the GTM calculation is an approximation and cannot be known to be optimal, it is theoretically possible for the SVM to show "poor" accuracy while still producing a denoised image exceeding the image quality of the denoised image produced from the GTM.

We therefore measure output image quality using the recently proposed quaternion structural similarity measure (QSSIM) by Kolaman and Yadid-Pecht, which, as mentioned previously, is an extension of the single-channel SSIM measure for colour images which has been shown to provide a more accurate measure of image quality

[9]. We also include quality measurements using PSNR, as this is a commonly used, classic metric; however, we would stress that since it has no perceptual basis, the QSSIM quality metric is considerably more meaningful.

Chapter 7

A Basic Two-Choice Classifier

In our first paper [3], we chose to classify between two candidate denoising algorithms: block-matching and 3D filtering (BM3D) and non-local means denoising (NLM). Both of these candidates were used with the recommended parameter values for a moderate level of denoising strength. Denoised ouput images therefore consist of pixels denoised with either NLM or BM3D, according to the classifier output.

Since this experiment served primarily as a proof of concept, noise levels for the training and test images were all kept constant at a VisualNoise level of 8, corresponding with a moderate but not overly destructive level of image noise.

For this classifier, mRMR found the following five features to be the most appropriate choices:

1. Local seam energy, Gaussian-blurred horizontal;

2. Global seam energy, vertical;

3. Image signature saliency;

4. Global seam energy, horizontal; and

5. Local colour variance.

These were therefore the features used as inputs to the SVM for training and testing.

7.1 Results

Quantitative image quality results for four test images may be found in Table 7.1. Additionally, visual results from two of these images are shown in Figure 7.1.

The results from this experiment clearly indicate that our algorithm outperforms both BM3D and NLM in many scenarios. Resulting output images were found to be both quantitatively superior – having generally higher QSSIM values – and visually superior to images denoised with either algorithm alone.

While the numerical results are convincing, it is the visual results that truly demonstrate the utility of this approach. In these two sample images, the outputs denoised by NLM preserve the image's sharper details (note especially the young man's facial features and the wire of the basket), but more uniform areas still exhibit large amounts of grainy noise (the young man's skin appears patchy, and the fridge is very noisy). In the BM3D denoised images, the opposite is true. Using SVMSID, however, the strengths of each are combined: the young man has both sharp features and smooth skin, while the fruit basket stands out sharply against a smooth fridge. The output of the classifier for each image may be seen in the last subfigure of each, with white pixels indicating it has selected NLM, and black indicating BM3D.

The SVM output maps and denoised images in Figure 7.1 may be compared with the GTM and "ideal" denoised output for these two images, which were previously shown in Figure 4.1.

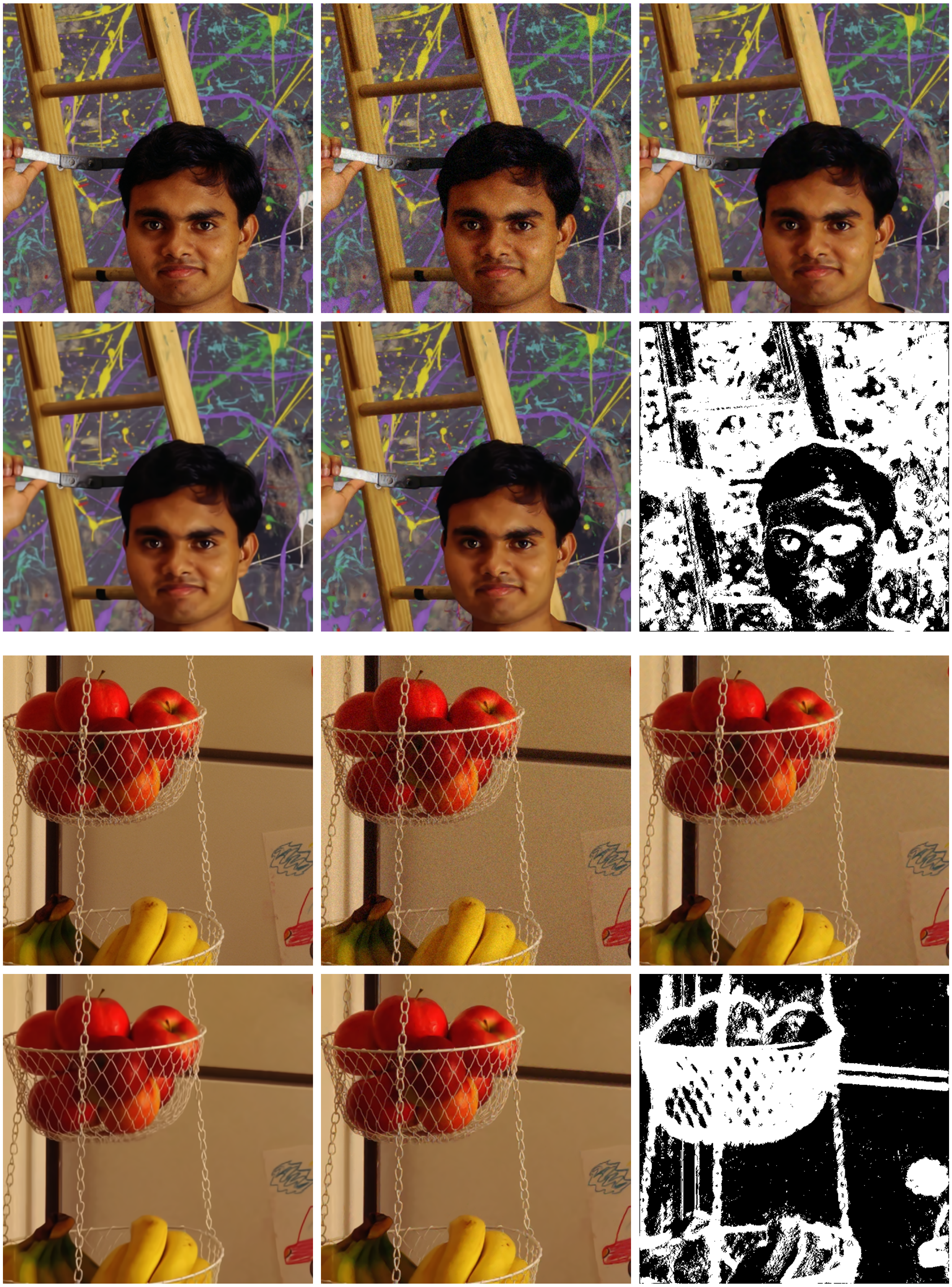


Figure 7.1: Denoising results for two test images (6.tif and 12.tif). *From top-left to bottom-right:* original image, noisy image, NLM denoising, BM3D denoising, our SVMSID denoising results, SVM classifier output (white = NLM, black = BM3D).

Image	Metric	Noisy	NLM	BM3D	SVMSID
3.tif	QSSIM	0.98080	0.97319	0.98514	**0.98613**
	PSNR	31.5680	30.2880	30.6969	**30.7151**
6.tif	QSSIM	0.98192	0.98265	0.98944	**0.99008**
	PSNR	31.4424	32.7573	**34.2852**	34.1402
10.tif	QSSIM	0.98410	0.99084	0.99210	**0.99283**
	PSNR	31.8306	33.8316	**35.2135**	35.1415
12.tif	QSSIM	0.98242	0.99242	0.99281	**0.99440**
	PSNR	31.4935	34.6790	35.2894	**35.5517**

Table 7.1: Image quality results for four images.

7.2 Computational Complexity

Before we proceed, it is worth noting that SVMSID's improved performance does not come without a substantial increase in computational complexity. The total computational cost of denoising an image using SVMSID is equal to the total time it takes for the following steps:

1. Calculating the values of each feature of interest for each pixel.
2. Performing SVM classification for each pixel.
3. Performing the appropriate denoising method on each pixel.

Note that we exclude ground truth map generation and the training of the SVM from this calculation; that is because training is only performed initially, and once a classifier model has been made, as many images can be classified with it as desired. We will briefly discuss the complexity of each of these components.

7.2.1 Feature Calculations

Most of the features used are block-based filters, and require a single series of calculations to be performed for each pixel ($O(n)$). The one notable exception to this is global seam energy; as the size of the image increases, not only does the number of seams to remove increase, but the amount of computations required to remove a single seam increases as well ($O(n \times \sqrt{n})$).

7.2.2 SVM Classification

SVM classification is by far the most computationally intensive portion of our method. The complexity of a support vector machine using an RBF kernel is $O(n_{sv})$, where n_{sv} is the number of support vectors used. The value of n_{sv} is dependant on many factors, with the most significant contributors being the cost parameter value C, the number of training points, and the number of features used. To classify an image of n pixels, the complexity would be $O(n \times n_{sv})$.

Empirically, classifying a 500×500 image during our various experiments took anywhere from around 5 minutes to several hours using a single core of assorted Core 2 Quad machines with between 4GB and 8GB of memory. For the results presented here, classification times of between 10 and 20 minutes per image would be more typical.

7.2.3 Candidate Denoising Methods

Finally, we must apply the desired denoising method to each pixel. For our purposes, we denoise the entire image using each method, and then select the appropriate pixels from each. Depending on the denoising candidate method, it may be possible to calculate outputs for only the desired pixels instead; however, any cost savings during this step would be minimal compared to the cost of the SVM classification.

7.2.4 Comparison with Traditional Methods

Certainly, the amount of time required to run SVMSID is much higher than traditional denoising methods. For this reason, SVSID is not intended to be a direct replacement for these methods. However, there are many cases when improved image quality is desired, regardless of the additional cost; it is for these applications that SVMSID is well suited.

Having demonstrated the strong performance of our approach between two denoising candidates and with a single level of image noise, and having briefly touched upon the computational complexity required, we will describe some extensions to our denoising method in the chapters that follow.

Chapter 8

A Two-Choice Classifier with Multiple Noise Levels

While the simple two-choice classifier discussed in the previous section displays promising results, there is a significant limitation to this first experiment: only a single, moderate level of noise was used for all images. For a denoising method to be considered truly useful, it should be able to perform well on images with a variety of different noise levels.

For these experiments, our improved local colour variance was included as a candidate feature. This was determined to have the highest classification relevance by mRMR, as well as significant redundancy with the previous first choice feature, Gaussian-blurred horizontal local seam energy. The six features selected for training and testing are therefore as follows:

1. Local colour variance (improved);

2. Image signature saliency;

3. Global seam energy, horizontal;
4. Global seam energy, vertical;
5. Local colour variance (original method); and
6. Local seam energy, vertical.

It is interesting to note that both colour variance methods were selected; while both were developed with the same motivation, their different approaches each lend the classifier a unique insight into the image structure.

8.1 Training Using A Single Noise Level

We begin by using our existing two-choice classifier, trained with images at a moderate noise level of VisualNoise = 8, and using it to classify test images with added VisualNoise levels of 2, 5, 8, 11, and 14, respectively.

While images with lower noise levels showed good performance, generally outperforming the output image quality of either NLM or BM3D alone, the classifier performance degrades significantly for the highest levels of noise. The classifier accuracy for 6 test images across each of the 6 VisualNoise levels, as well as the QSSIM for each of these ouput images, is shown in Figure 8.2.

The decrease in output QSSIM for images with higher noise levels is partially a reflection of the decreased performance of both of our candidate denoising algorithms for higher noise levels. However, this is not the full story, as classifier accuracy is also considerably diminished.

This is not unexpected, as the values of the features used to classify each pixel of the test images will certainly vary with the level of noise present. Nevertheless, this suggests that classification accuracy may be unduly compromised simply because we do not not have a representative enough selection of training points. With this in mind, we perform a second experiment using a modified training set.

8.2 Training Using Multiple Noise Levels

In an attempt to improve the denoising performance for higher noise levels, we modified the selection criteria for training points to better represent the variety of test images. That is, training points for this experiment were not simply from images of a single noise level, but of a wide variety of noise levels (VisualNoise levels of 2, 5, 8, 11, and 14).

Since the values of each image feature can logically be expected to vary more with increasing noise levels, we select a greater number of training points from images with higher noise levels.

The testing results using this classifier may be seen in Figure 8.3. Both the classifier accuracy and the QSSIM of output images are considerably improved for images with high noise levels, while remaining similar to before for images with low levels of noise. Overall, this classifier behaves much more robustly across a wide range of noise values.

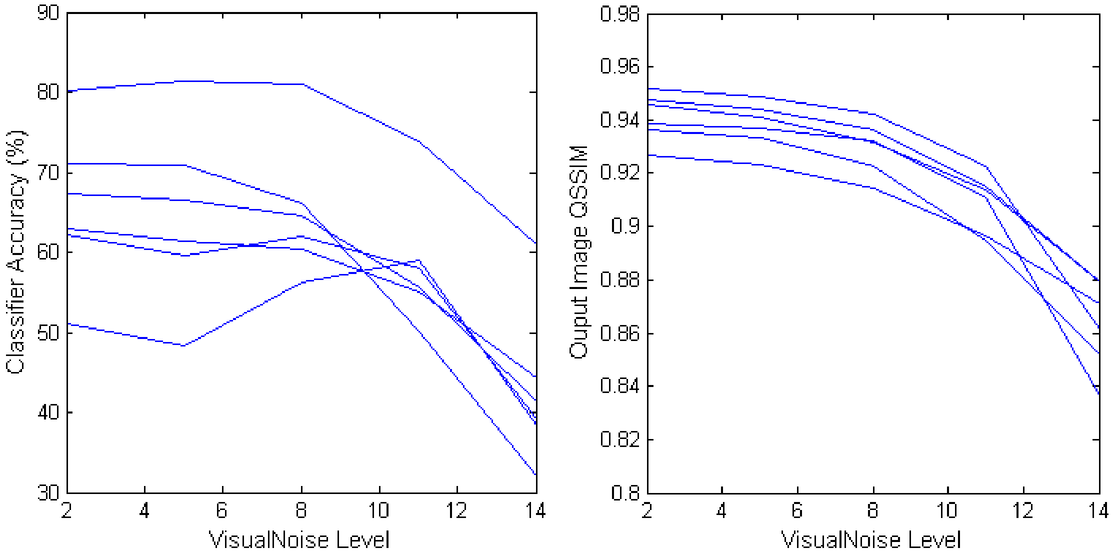


Figure 8.2: The resulting classifier accuracy and QSSIM when using a classifier trained using only images with a single, moderate level of noise (VisualNoise = 8). Each line represents the performance for different noise levels added to the same test image.

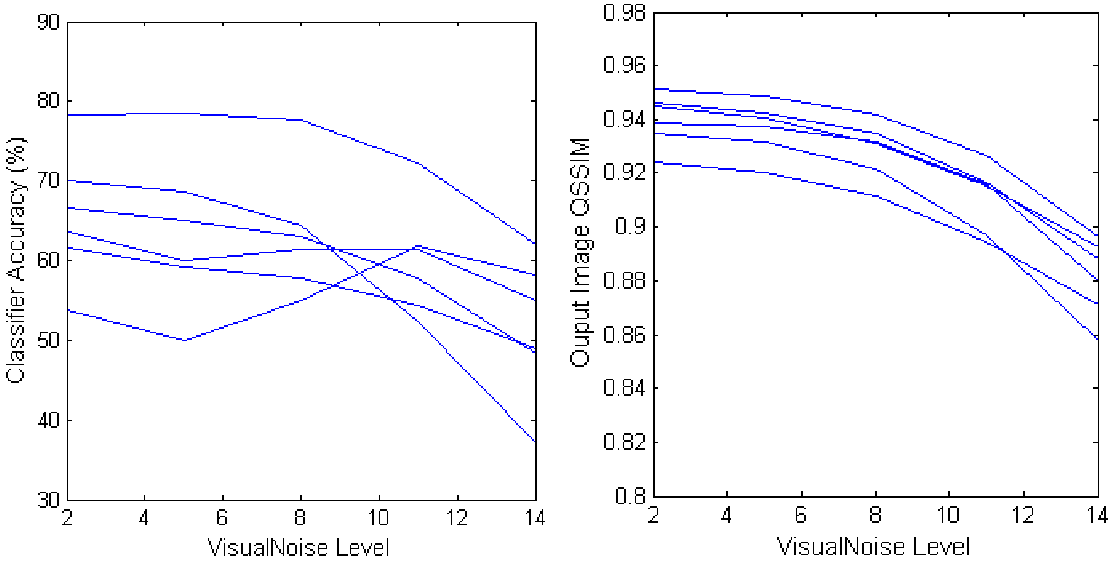


Figure 8.3: The resulting classifier accuracy and QSSIM when using a classifier trained using images with a variety of noise levels (VisualNoise = 2 – 14). As before, each line represents the performance for different noise levels added to the same test image.

Chapter 9

Proposed Extensions

We will now briefly discuss some further extensions to SVMSID that are currently being investigated for inclusion in an upcoming journal publication.

9.1 Classification Between Multiple Parameters

Few denoising algorithms are parameterless, and by changing the parameters used with these algorithms, we can significantly alter the strength and quality of their denoising performance. Thus far, SVMSID has been used to select between different denoising algorithms; it makes sense that it could also be used to select which parameter value of a single algorithm is best suited to each area of an image.

For our experiments, four different denoising parameters are used for NLM – and, an a separate experiment, for BM3D. These were selected to range from barely perceptible denoising to denoising levels that are nearly destructively strong.

Preliminary results from these experiments has produced output images which are generally of slightly lower quality than simply denoising the entire image using the best of the four candidate parameters alone. It should be noted that this, in and of itself, is not necessarily a failure; given no other information about which denoising parameter to use on an image, SVMSID would still perform better than chance. However, there remains room for improvement; classifier accuracy is also slightly lower than expected, considering the performance of the two-choice classifier.

The main reason for this performance gap is likely due to the training point selection method. With a larger number of classes, the classification boundaries become considerably more subtle, and selecting a good sample of training points becomes more difficult. Experiments are currently being run to determine the best point selection method for this classifier; and, likely, the insights gained here will be useful for improving classification performance with the two-choice classifier as well.

9.2 Hierarchical Classifier

As a logical extension of both the two-choice classifier and a classifier between multiple parameters, we are also performing experiments using a hierarchical classifier. In this method, we use multiple support vector machines to select first the algorithm to use for each pixel, and then the parameter with which to apply it. Because this mechanism both selects and tunes the denoising algorithm for each pixel, this hierarchical structure should be able to significantly improve the quality of output images.

Chapter 10

Conclusions and Further Work

As we have demonstrated, the use of a support vector machine classifier with various image features shows strong promise in improving upon existing denoising algorithm performance.

While SVMSID already produces favourable results, there remains considerable room for further development. In addition to our proposed extensions, performance could be improved with the development of additional image feature choices and the addition of other denoising candidate classes. This versatility is perhaps SVMSID's greatest strength; should a new denoising algorithm be developed, SVMSID is not rendered obsolete, but rather it can use this new algorithm as a denoising candidate to further improve its performance.

One significant consideration, however, is efficiency; support vector machines are necessarily slow and computationally intensive to run. We plan to investigate methods of grouping pixels into blocks or superpixels for classification to significantly decrease the amount of time required to classify each test image.

Bibliography

[1] K Dabov, A. Foi, V. Katkovnik, and K. Egiazarian, "Image denoising by sparse 3-D transform-domain collaborative filtering," *IEEE Transactions on Image Processing: a Publication of the IEEE Signal Processing Society*, vol. 20, no. 1, pp. 2080 – 2095, Jan. 2007.

[2] Antoni Buades, Bartomeu Coll, and J. M. Morel, "A non-local algorithm for image denoising," *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2, pp. 60–65, 2005.

[3] Laura McCrackin and Shahram Shirani, "Strategic image denoising using a support vector machine with seam energy and saliency features," in *International Conference on Image Processing (ICIP)*, 2014, pp. 2684–2688.

[4] Lei Zhang, Xiaolin Wu, Antoni Buades, and Xin Li, "Color demosaicking by local directional interpolation and nonlocal, adaptive thresholding," *Journal of Electronic Imaging*, vol. 20, no. 2, 2011.

[5] Suk Hwan Lim, "Characterization of noise in digital photographs for image processing," in *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*, 2006, vol. 6069, pp. 219–228.

[6] Henrique S. Malvar, Li-Wei He, and Ross Cutler, "High-quality linear interpolation for demosaicing of bayer-patterned color images," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2004, vol. 3, pp. iii–485–8.

[7] International Standards Organization (ISO), "Photography – electronic still-picture imaging – noise measurements," ISO 15739:2013, International Organization for Standardization, Geneva, Switzerland, 2013.

[8] Zhou Wang, Alan Conrad Bovik, Hamid Rahim Sheikh, and Eero P. Simoncelli, "Image quality assessment: from error visibility to structural similarity.," *IEEE Transactions on Image Processing: a Publication of the IEEE Signal Processing Society*, vol. 13, no. 4, pp. 600–12, 2004.

[9] Amir Kolaman and Orly Yadid-Pecht, "Quaternion structural similarity: a new quality index for color images.," *IEEE Transactions on Image Processing: a Publication of the IEEE Signal Processing Society*, vol. 21, no. 4, pp. 1526–36, Apr. 2012.

[10] Shai Avidan and Ariel Shamir, "Seam carving for content-aware image resizing," in *ACM Transactions on Graphics (TOG)*. ACM, 2007, vol. 26, p. 10.

[11] Rafael C. Gonzalez and Richard E. Woods, *Digital Image Processing (3rd Edition)*, Prentice Hall, Upper Saddle River, New Jersey, 2006.

[12] K. McLaren, "XIII – The Development of the CIE 1976 (L* a* b*) Uniform Colour Space and Colour-difference Formula," *Journal of the Society of Dyers and Colourists*, vol. 92, no. 9, pp. 338–341, 1976.

[13] Roy Berns, *Billmeyer and Saltzman's Principles of Color Technology*, Wiley, New York, 2000.

[14] Brian W Keelan, Elaine W Jin, and Sergey Prokushkin, "Development of a perceptually calibrated objective metric of noise," in *IS&T/SPIE Electronic Imaging*. International Society for Optics and Photonics, 2011, pp. 786707–786707.

[15] Xiaodi Hou, Jonathan Harel, and Christof Koch, "Image signature: highlighting sparse salient regions.," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 1, pp. 194–201, July 2011.

[16] H. Peng, Fulmi Long, and C. Ding, "Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 27, no. 8, pp. 1226–1238, Aug 2005.

[17] Chih-Chung Chang and Chih-Jen Lin, "LIBSVM: A library for support vector machines," *ACM Transactions on Intelligent Systems and Technology*, vol. 2, pp. 27:1–27:27, 2011.

[18] Simon Haykin, *Neural Networks and Learning Machines (3rd Edition)*, Prentice Hall, Upper Saddle River, New Jersey, 2008.

[19] Chih-Wei Hsu, Chih-Chung Chang, Chih-Jen Lin, et al., "A practical guide to support vector classification," 2003.