4TN4 Project: A novel denoising approach

Anonymous CVPR submission

Paper ID 400113347

Abstract

This paper is a proposed noise reduction process in grayscale images using an SVR machine learning model. This proposed process consists of training a model based on the pixel and edge information surrounding a pixel, and estimating the best pixel value for each pixel in an image. The approach works well to reduce noise while preserving edge information, however, there are occasional artifacts along some edges in the final image.

1. Introduction

All images contain some noise and it is almost always desirable for there to be less noise in an image. There are many techniques that can be used to reduce noise in images from increasing the signal to noise ratio at the time of capture, to post-processing.

1.1. Sources of Noise

Noise is a visible distortion in the image due to random changes in the intensity of the image [2]. The source of noise can be due to thermal activity, electromagnetic activity, or various other physical phenomena at the time of image capture [2].

1.2. Time of Capture Noise Reduction

The best way to reduce noise in an image is to never introduce it in the first place - in other words, keeping the signal to noise ratio (SNR) high. This can be done at time of capture by increasing exposure time, thus collecting more samples and inherently increasing the SNR. Consider Figures 1 & 2: when the exposure time is longer, a clearer image is produced. This method obviously can not be used when a longer exposure time causes motion blur in the image from moving objects, which can include the photographer's body, when taking the image with a handheld device without some method for stabilization.

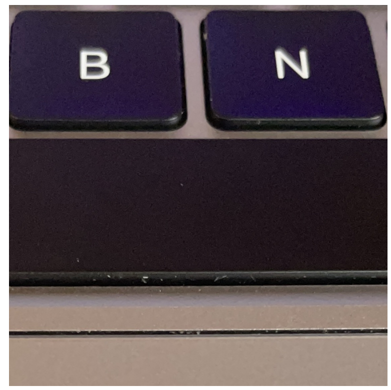


Figure 1. Image taken with 1/60s exposure time

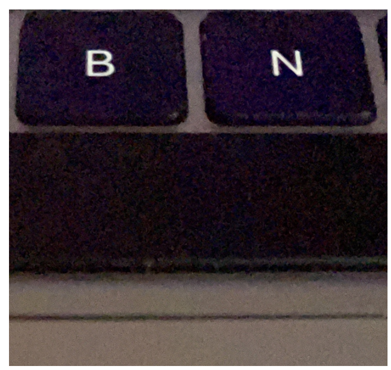


Figure 2. Image taken with 1/1006s exposure time

1.3. Time of Capture Noise Reduction with Post Processing

There are also techniques to increase the SNR of an image by taking many images with lower SNRs and then using various techniques to stitch the images together, and average the information gathered in areas that should stay constant across each image. This is a simpler problem when an image is taken of a static scene using a method to ensure the camera does not move between image captures, in which case, each pixel in the final image can simply be calculated by taking the average of all of the noisy images.

1.4. Traditional Techniques

Two of the simplest traditional techniques for noise reduction are mean and median filtering.

1.4.1 Mean Filter

A mean filter works by replacing a pixel's value with the average of its N surrounding pixels. This will result in a reduction of noise, but, a loss of detail as well.

The calculation for a mean filter is as follows:

$$p' = \sum_{n=0}^{N-1} \frac{p_n}{N}$$

where p' is the new pixel value, p_n are the pixels surrounding the central pixel, and p_0 is the central pixel.

1.4.2 Median Filter

Noise reduction by median filtering works best for salt and pepper noise, in which sharp and sudden changes in the image result in black and white dots in the image [2]. The median filter replaces a pixel's value with the median of its surrounding pixels. In some cases this is better than a mean filter because a few outlier pixel values don't significantly change the result.

1.5. SVR

Support Vector Regression (SVR) is a machine learning (ML) technique used to regress data and is the regressive equivalent of the support vector machine (SVM) classifier. Consider a feature set X with k features, where each feature in X has N dimensions and has a corresponding result in y. A SVM classifier works by mapping each data-point in euclidean space in N dimensions and finding the line that separates all of the like-classified data-points, by reducing the error across the data-set.

The error across the data-set can be expressed as:

$$error = \sum_{i=0}^{k} (\tilde{y}_i - y_i)^2$$

Where $\tilde{y}$ is the SVM generated estimate of the output given the input feature-set.

Once the model is trained, it will output an estimate class given an input feature set which will ideally be the correct class most of the time.

SVR is very similar to SVM except instead of discrete classes into which the output for a given input is classified, the output can be in a continuous range of values. This is useful for approximating pixel values when de-noising, since the goal is to determine the best value for a pixel, not to classify it.

2. Proposed SVR approach to denoising

The proposed method for reducing the noise in a noisy image is to train an SVR model to predict the value of each individual pixel in the noisy image given its surrounding pixel values. To mitigate detail being lost around edges as with a mean filter method, edge information will also be included in the training set to promote a different noise reduction strategy near edges. The training set consists of the SIDD Small data-set [1] compiled by researchers at York University, which includes over 100 images captured on various smartphones. Each image has two versions: a so called "ground truth" image which is an image with very little noise, and a noisy image, which the same image as the ground truth image but with added noise. This can be more succinctly put by:

$$f = g + n$$

where f is the final noisy image, g is the ground truth image, and n is the noise in the image. Our goal is to try and recover g from f.

The SIDD data-set will be split into an 80-20 split for training and testing respectively.

2.1. Training

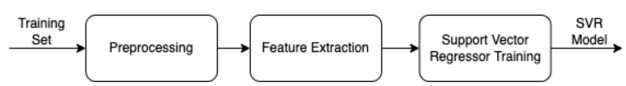


Figure 3. Model Training Flow Diagram

Figure 3 outlines the model training workflow.

2.1.1 Preprocessing

Each noisy image and ground-truth image will be preprocessed before feature extraction. The noisy images will each be filtered with a Gaussian filter to smooth the image and reduce the noise, and improve the extraction of features. Following the Gaussian filter, histogram equalization will be utilized on certain images to improve the contrast, and improve edge detection in the feature extraction stage as required. The ground truth image's histogram will be equalized as well to match the noisy image, in order to improve the pixel estimation in the model.

2.1.2 Feature Extraction

As described above, the features that will be used for each pixel estimate are the surrounding pixels of the central pixel, and the edge information of the input image. Therefore, following preprocessing, Canny edge detection will be run on the image to extract the edge information, which will be stored and henceforth refer to as the "edge image".

CVPR #400113347

Consider two images:

0	0	1	0
1	1	1	1
0	1	1	0
0	0	1	0

Table 1. Preprocessed noisy image

0	0	1	0
1	1	1	1
0	0	1	0
0	0	1	0

Table 2. Edge image

note that the edge image contains only the edge information, so the pixel at (1,2) in the noisy image (Table 1), becomes 0 in the edge image, which is denoted by the red colored pixel in Table 2 (index counting starts from 0 in the top left corner).

Given these images, the features desired for extraction are the 3x3 pixel matrix surrounding the estimated pixel and the 3x3 edge matrix surrounding the estimated pixel. For example if estimating the pixel at (1,2), the two matrices extracted would be:

1	1	1
0	1	1
0	0	1

Table 3. Pixel information surrounding estimated pixel

1	1	1
0	0	1
0	0	1

Table 4. Edge information surrounding estimated pixel

Following the extraction of the pixel and edge matrices, the data is flattened into an array of length 18, in order to make the data more suitable for SVR training. Each matrix is flattened such that a matrix such as:

1	2	3
4	5	6
7	8	9

would become the array: $[1, 2, 3, 4, 5, 6, 7, 8, 9]$.

For example, if estimating the pixel at (1,2), the entire feature-set X would be: $[1, 1, 1, 0, 1, 1, 0, 0, 1, 1, 1, 1, 0, 0, 1, 0, 0, 1]$.

2.1.3 SVR Training

Extracting all of the features mentioned above for each pixel in every image is more data than needed. Only around 10000 data-points will be used to train the model, therefore, extracting the features for every 100-400 pixels (by skipping 10-20 pixels in both x and y directions) in each image will still yield far more data than required. To procure the best training data, the features from only 1/100 pixels in an image will be extracted, and then that large data-set will be shuffled with some deterministic algorithm with a fixed kernel, by manually selecting the most appropriate kernel.

As mentioned above, 80% of the 10000 data-point data-set will be used for training the SVR model, while 20% will be used for testing it.

So, the feature set outlined above, X, and the corresponding ground truth data, y, will be used to train the SVR model. The model will be fit to minimize the error given the error function described above. Note that y is simply the ground truth pixel value for the corresponding noisy pixel in the input image.

2.2. Noise reduction process

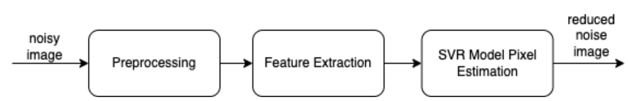


Figure 4. Noise Reduction Flow Diagram

Figure 4 outlines the noise reduction workflow.

2.2.1 Preprocessing

The preprocessing stage for the proposed noise reduction method is very similar to the preprocessing stage in the proposed training method outlined above, with one key difference: histogram equalization will not be not applied to the input image, since the goal of the noise reduction process is only to remove noise, not to change any other properties of the image.

Therefore, a copy of the input image will be made, which will then be filtered with a Gaussian filter, and on which histogram equalization will be applied.

2.2.2 Feature Extraction

As in the training process, Canny edge detection will be performed on the processed input image to acquire the edge image. Then the surrounding pixel information of the unprocessed input image and the edge image for the estimated pixel will be extracted and flattened into an array.

2.2.3 Pixel Estimation

The pixel estimation process will iterate over each pixel in the image to be denoised, and will extract the aforementioned feature-set and using the trained SVR model will estimate the value of the pixel. Each pixel value in the image will be replaced by the value of the estimated pixel.

3. Results

3.1. SVR Model Training

Following the model training process described above, a data-set of 10000 data-points was extracted from the SIDD Small data-set [1]. 80% of which was used to fit the SVR model and 20% of which were used to test the model.

For a given sample image such as the one shown in Figure 5(a) the output after filtering with a Gaussian filter is shown in Figure 5(b).

(a) Original Input Image (b) Gaussian Filtered Image

Figure 5. Image before and after Gaussian filtering

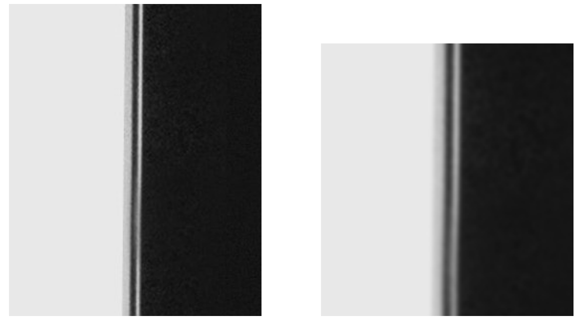
Figure 5(b) is subtly different than Figure 5(a), there is a slight blur effect on the image that aids in edge detection in the following step and is more noticeable in the blown up comparison in Figure 6.

Following the Gaussian filtering step, Canny edge detection was run on the image resulting in the image shown in Figure 7.

Once the edge image was acquired, the feature-set X was extracted as described above. The same procedure was repeated with many images and the best model fit was found to have an accuracy of 96.18%.

3.2. Noise Reduction

When an image was denoised the process was executed as described above: the input image was filtered, the edge image was acquired, and the pixel values were estimated based on the extracted features.



(a) Original Input Image Blown Up (b) Gaussian Filtered Image Blown Up

Figure 6. Image before and after Gaussian filtering close up

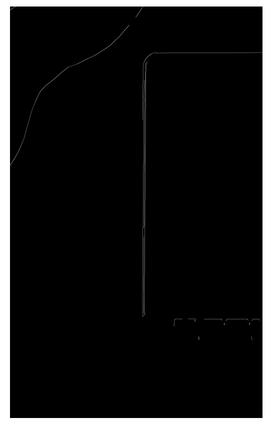


Figure 7. Sample input image after Gaussian filtering

For example, for the input image shown in Figure 8, the edges detected are shown in Figure 9, and following a pixel estimation process for each pixel in the image the resulting image shown in Figure 10(b) shows reduced noise compared to the input image. When the proposed pixel estimation technique is compared to the mean filter noise reduction approach, shown in Figure 10(c), it is evident that the pixel estimation approach provided better noise reduction and a smoother looking image.

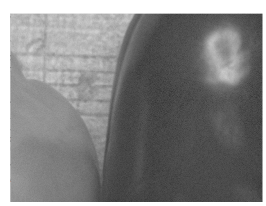
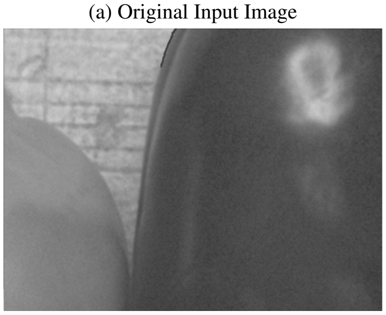


Figure 8. Sample image inputted for noise reduction

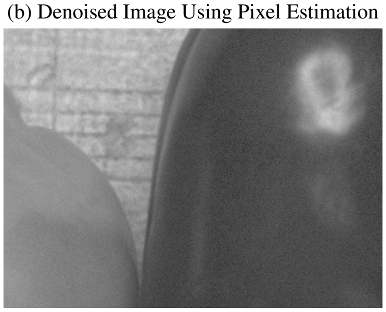
Table 5 shows some data regarding the sum of the square error in two different images, one of peppers, and the other of whiteboard markers. PE error refers to the error of the proposed pixel estimation approach, MF error refers to the

Figure 9. Sample edge image

(a) Original Input Image



(b) Denoised Image Using Pixel Estimation



(c) Denoised Image Using Mean Filtering

Figure 10. Denoising techniques compared with input

error of a simple mean filter approach to noise reduction, and noise error refers to the error of the raw noisy input image compared to the ground truth image.

Image	PE Error	MF Error	Noise Error
Peppers	91545895	94293931	94723484
Markers	690461178	693005449	692915463

Table 5. Sum of square error in images with different noise reduction approaches

As evident from the data presented above in Table 5, the PE approach offers an improvement over both the noisy image and the mean filtered image.

3.3. Limitations

As seen in the top of the image shown in Figure 10(b), there is an artifact along the edge of the larger object in the image. This is an issue that presented itself in many images; unfortunately the cause is unclear. Selecting better data-points when training (perhaps ones that include more edge information) could yield better results.

References

[1] Abdelrahman Abdelhamed, Stephen Lin, and Michael S. Brown. A high-quality denoising dataset for smartphone cameras. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018. 2, 4

[2] Rohit Verma. A comparative study of various types of image noise and efficient noise removal techniques, 2013. 1, 2