Math 4900 Project I Summary (Part I)

Lai Chun Yin (1155125168)
Lui Him (1155127647)

Contents

1. Introduction

Image Denoising is an important topic in image processing. The goal of image denoising is to reduce noise in an image. There are some applications for image denoising, for example, it can be used to make an image look better and it can be used to preprocess an image for some other image processing tasks such as image segmentation and image classification.

One of the earliest approaches to the denoising problem was to solve the following minimization problem:

$$\min_f \ \frac{1}{2}\int\int_\Omega |f-g|^2\,dx\,dy \quad +\lambda\int\int_\Omega |\nabla f|^2\,dx\,dy \qquad (1)$$

, where g is the noisy image, f is the image to be solved and $\lambda > 0$ is a parameter. The first term is called the fidelity term that ensure that the denoised image would not be “too different” from the original image and the second term is called the smoothing term (or the regularization term) that is used to reduce the oscillation of the image.

However, the above model is not a good one because the model penalizes too much the edges, making the model non-edge preserving.

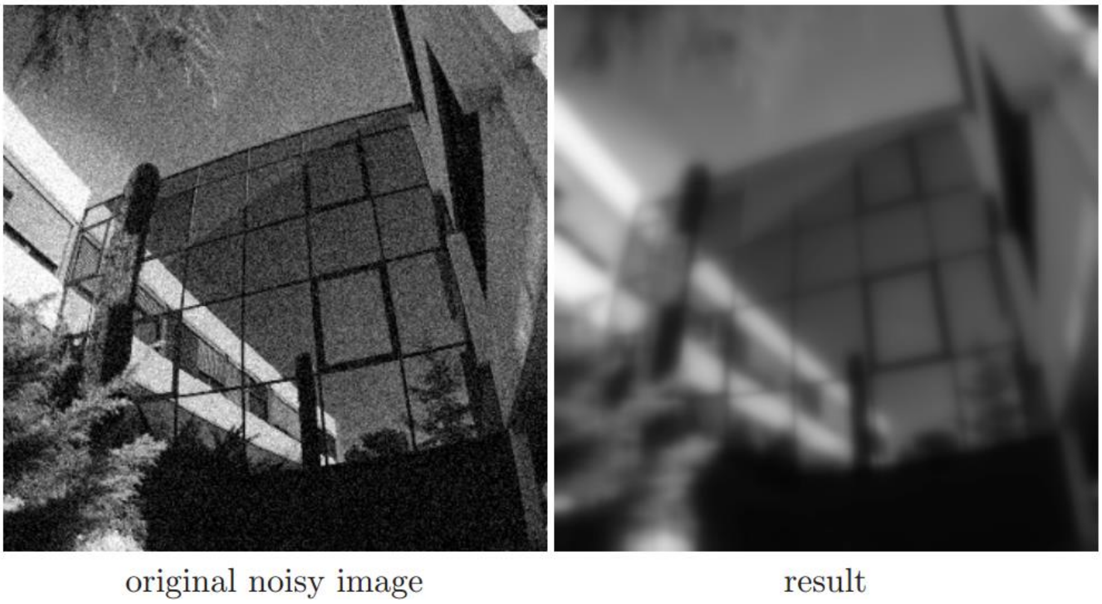


Fig 0.1

Therefore, in 1992, Rudin, Osher and Fatemi proposed the Total Variation Model (TV model / ROF model), which solves the following minimization problem:

$$\min_f \{\frac{1}{2}||f - g||_2^2 + \lambda\, ||\nabla f||_1\}$$

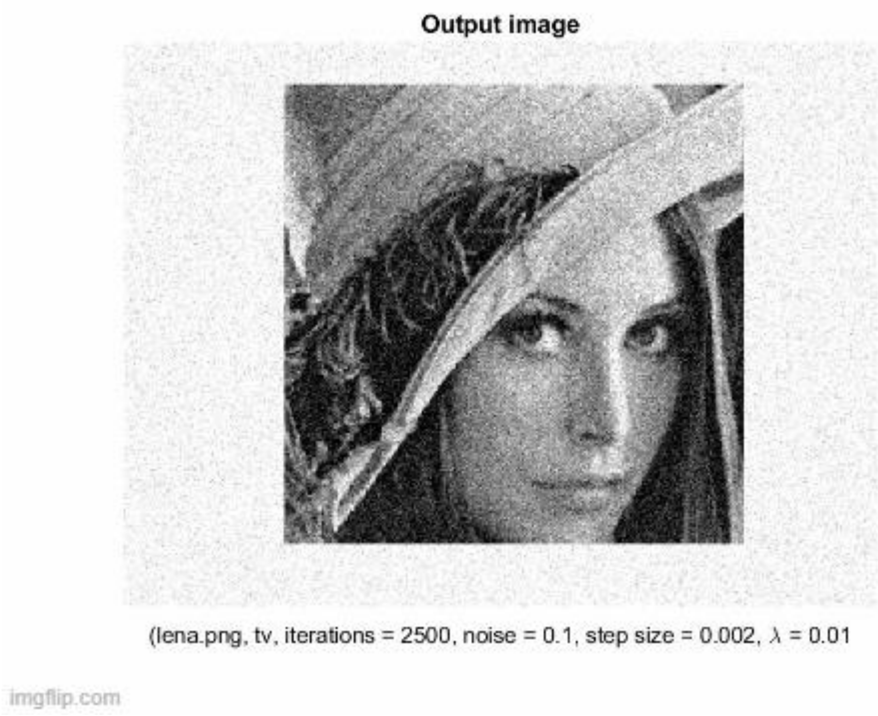
or equivalently,

$$\frac{1}{2} \int\int_{\Omega} (f(x,y) - g(x,y))^2 \, dx\, dy + \lambda \int\int_{\Omega} |\nabla f|(x,y) dx\, dy \qquad (2)$$

, where g is the noisy image, f is the image to be solved and $\lambda > 0$ is a parameter. The difference between the TV model (2) and the first model (1) is that, TV model minimizes the L1-norm of the gradient of f while model (1) minimize the L2-norm of the gradient of f. Although the TV model is just a slight modification of model (1), the TV model is much better than model (1) in terms of preserving edges.

2. Role of the parameter λ

The parameter λ in the second term of the TV model is used to control the smoothness of the denoised image. Larger λ gives a smoother but blurrier image and smaller λ gives a less blurry image but the noise may not be completely removed. Below is an animation that shows the outputs of the TV model at different λ,



Gif 0.1

3. <u>Performance of the TV model</u>

Now, let us compare the images before and after doing TV denoising.

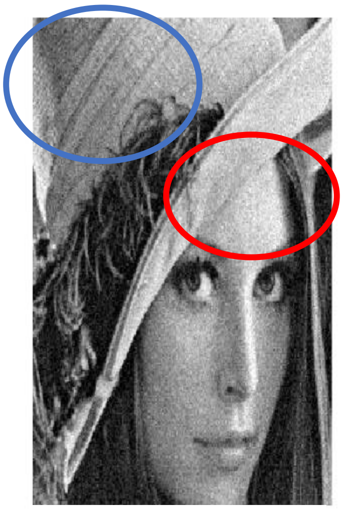


Fig 0.2. Noisy image

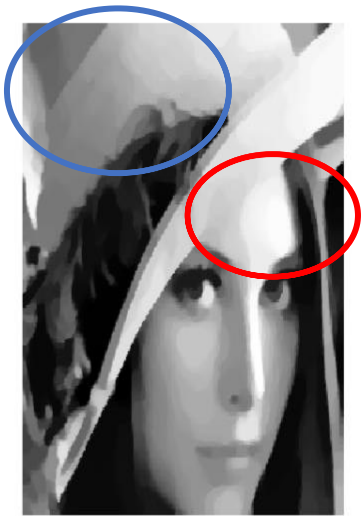


Fig 0.3. Image denoised using TV model

In Fig 0.2 and 0.3, we can see that the TV model has good denoising ability and, unlike the previous model, the TV model can preserve the edges at the same time. However, the TV model is not perfect. The model has some drawbacks. First, the denoised image loses some texture. For example, in Fig 0.2, we can clearly see some stripes on the lady's hat, but after denoising it using the TV model, we see that the stripes are smoothed out. (see the blue circles)

Second, there are some "fake edges", which do not exist in the original image, on the denoised image. These "fake edges" are enclosing different regions of the image, forming different color patches in the image with distinct colors. This makes the denoised image looks like an oil painting instead of a real image. For example, we can see the forehead of the lady in Fig 0.2 is smooth. However, in Fig 0.3, we see that the forehead of the lady consists of roughly 3 regions with distinct colors. (see the red circles)

If we plot the pixel values over the image domain, we will see that the plot looks like staircase. This is a famous drawback of the TV model known as the "staircase effect".

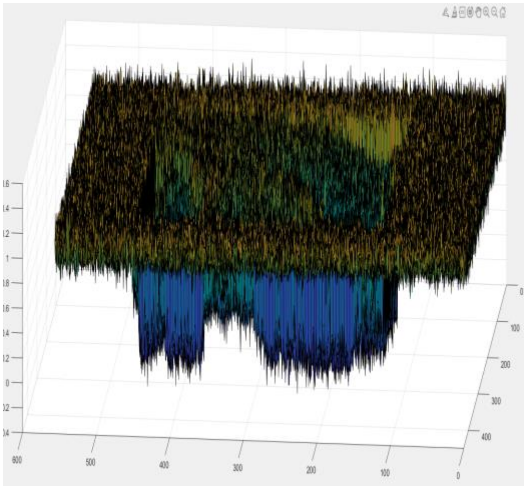


Fig 0.4 Pixel values of a noisy image

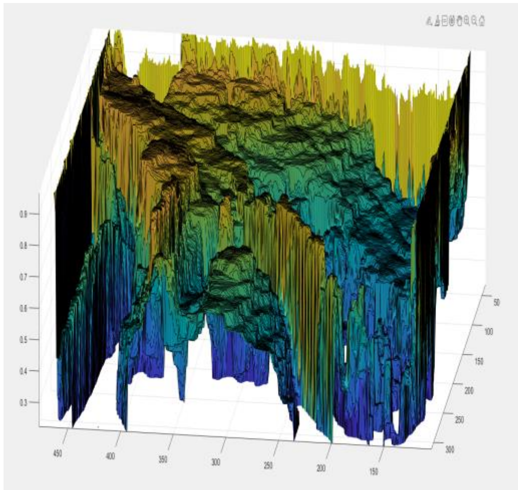


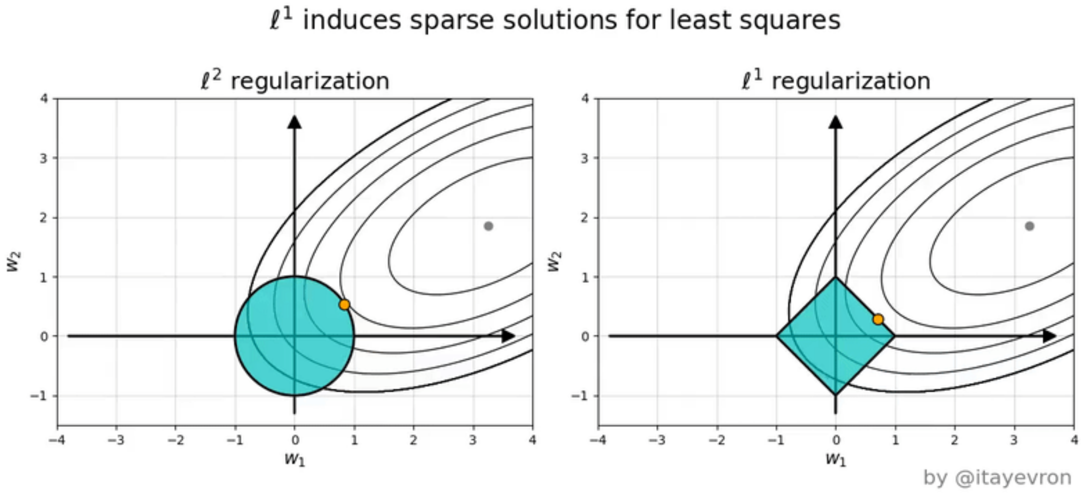
Fig 0.5 Pixel values of a TV-denoised image

The aim of our project is to construct a smoothing term that can reduce the staircase effect produced by the TV model.

4. <u>What causes the staircase effect?</u>

Before looking into the proposed models, we first need to investigate what causes the staircase effect. Mathematically, the staircase effect means that the denoised image is a piecewise constant image. This is because the TV model regularizes the magnitude of the gradient with respect to the L1 norm. L1 regularization produces sparse solution, that is, a solution that contains many zero elements.

The reason for this is that the unit sphere under the L1 norm has a "diamond shape" with its corners located on the axis, that is, at the corner, only 1 element is nonzero. For example, consider a simple 2-dimensional case. Suppose a quadratic function is now being minimized. From the animation below, it can be seen that the optimal point occurs at the corners, that is, one of (1, 0), (-1, 0), (0, 1) and (0, -1). Note that in any of these 4 cases, only one element is nonzero.



Gif 0.2

In an optimization problem, the optimal point usually occurs at one of these corner points. This explains why the optimal solution is sparse.

In our problem, the regularization term is $||\nabla f||_1$, therefore, the solution would be an image whose gradient magnitude is zero at many pixels. Since the gradient magnitude at a point represents how much do the pixel values oscillate near that point, if it is zero, then that means the pixel values do not change near that point. This explains why the TV-denoised image is a piecewise constant image.

5. <u>A Modified version of the TV model</u>

Now, consider a modified version of the TV model,

$$\min_u \{ \frac{1}{2}||u - f||_2^2 + \lambda \, || \, g\nabla u \, ||_1 \} \qquad (3)$$

, where f is the noisy image, $g = \frac{1}{1+|\nabla(G_\sigma * f)|^2}$ and G_σ is a gaussian kernel with standard deviation σ.

In model 3, the function g acts as an edge detector. To see its function as an edge detector, we shall investigate g in detail. First, the constant 1 in the denominator of g is used to avoid singularity caused when $|\nabla(G_\sigma * f)|^2 = 0$ Second, in the denominator, there is a term $|\nabla(G_\sigma * f)|^2$.

Note that the convolution between a Gaussian kernel and an image means that Gaussian filtering is applied on the image. Therefore, the term $G_\sigma * f$ represents a smoothed image. Then, the gradient magnitude of $G_\sigma * f$ will be small at location where the corresponding location of f is an interior point or noise, because $G_\sigma * f$ is an image, whose noise is smoothed out. And the gradient magnitude of $G_\sigma * f$ will be large at the location where the corresponding location of f is an edge because if a suitable σ is chosen, the edges in f will not be completely smoothed out in $G_\sigma * f$. And, by the definition of g, we see that if $|\nabla(G_\sigma * f)|^2$ large, then the value of g will be small and vice versa.

The cases can be summarized in the following table,

	$\lvert\nabla u\rvert$	$\lvert\nabla(G_\sigma * f)\rvert^2$	g
Noise	High	Small	High
Edge	High	High	Small
Interior points	Small	Small	High

Note that in an image, both the noise and the edges have high frequencies or high gradient magnitude, that is $|\nabla u|$ is large. However, from the table above, we can see that the value of the regularization term $|| g\nabla u ||_1$ is large only when the image u contains much noise. If a point in u corresponds to an edge, although $|\nabla u|$ at that point will be high, the value of g at that point will be small, so $|| g\nabla u ||_1$ will not be too large at that point. That means only high frequency parts that correspond to noise will be penalized, and hence the edges and details can be preserved better. This explains why the function g acts as an edge detector in model (3).

Below are some examples,

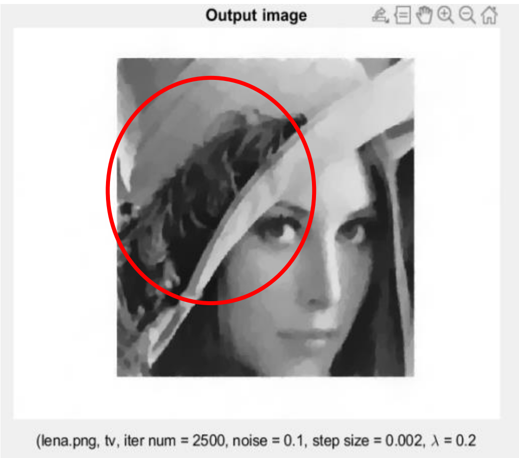


Fig 0.6

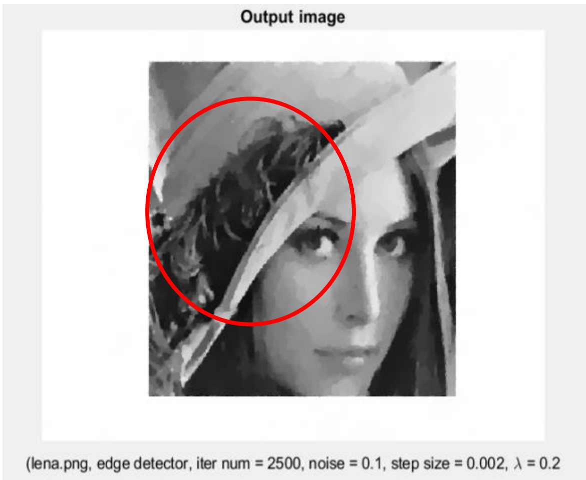


Fig 0.7

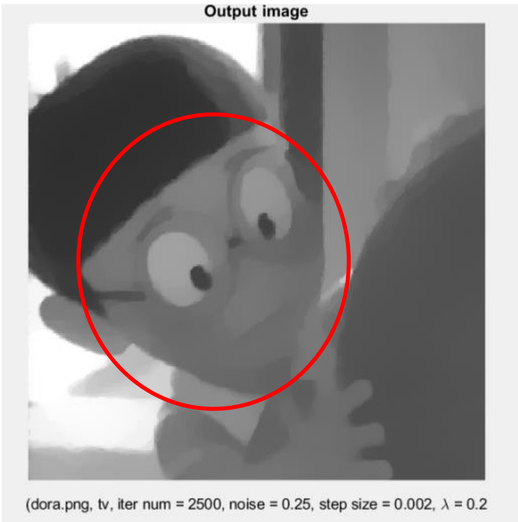


Fig 0.8

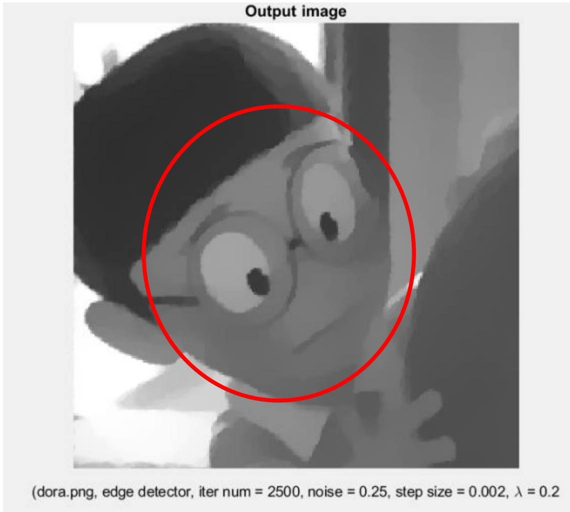


Fig 0.9

Fig 0.6 and Fig 0.8 are the results of the ordinary TV model while Fig 0.7 and Fig 0.9 are the results of the TV model with the edge detector g under the same λ. We can see that in Fig 0.7, the edges of the feather on the lady's hat are much sharper compared to Fig 0.6. And in Fig 0.9, we can clearly see the boy's glasses, eyebrows, and mouth, but in Fig 0.8, these details are so light that they almost disappear. This illustrates that, by adding the edge detector g, the details and edges can be preserved better in the images.

6. <u>Numerical Method used</u>

Recall that in the TV model, we are minimizing

$$J(f) = \frac{1}{2} \int\int_{\Omega} (f(x,y) - g(x,y))^2 \, dx \, dy + \lambda \int\int_{\Omega} |\nabla f|(x,y) dx \, dy$$

Define

$$L(x, y, f, f_x, f_y) = \frac{1}{2}\left(f(x,y) - g(x,y)\right)^2 + \lambda |\nabla f|(x,y)$$

Then,

$$J(f) = \int\int_{\Omega} L(x, y, f, f_x, f_y) \, dx \, dy$$

Since the J(f) is convex, gradient descent method could be applied to obtain the global minimum.

The iteration scheme is given as follows,

<u>Step 1:</u>

Let the noisy image be the initial guess.

<u>Step 2:</u>

At step $n \geq 0$, for all pixels, the image f is updated using the following formula,

$$f^{n+1} = f^n - t \, \nabla J(f^n)$$

, where t is the time step (t is a constant for convenience).

The gradient of J(f) can be found by finding the Euler-Lagrange equation of J. The Euler-Lagrange equation of J is given by

$$\frac{\partial L}{\partial f} - \frac{\partial}{\partial x}\frac{\partial L}{\partial f_x} - \frac{\partial}{\partial y}\frac{\partial L}{\partial f_y} = 0$$

Note that

$$\frac{\partial L}{\partial f} = f - g$$

$$\frac{\partial L}{\partial f_x} = \lambda\frac{f_x}{\sqrt{f_x^2 + f_y^2}}$$

And

$$\frac{\partial L}{\partial f_y} = \lambda\frac{f_y}{\sqrt{f_x^2 + f_y^2}}$$

Therefore,

$$\frac{\partial L}{\partial f} - \frac{\partial}{\partial x}\frac{\partial L}{\partial f_x} - \frac{\partial}{\partial y}\frac{\partial L}{\partial f_y}$$

$$= (f - g) - \lambda\nabla \cdot \left(\frac{\nabla f}{|\nabla f|}\right)$$

Hence,

$$-\nabla J(f) = -EulerLagrangeEquation$$

$$= -(f - g) + \lambda\nabla \cdot \left(\frac{\nabla f}{|\nabla f|}\right)$$

Since the above formula is continuous, it is discretized using the following finite difference scheme so that it can be computed using computer.

The x and y components of the gradient of a two-dimensional function f are approximated as

$$(\nabla \mathrm{f})_{\mathrm{x}} \approx f(x+1, y) - f(x, y)$$

and

$$(\nabla \mathrm{f})_{\mathrm{y}} \approx f(x, y+1) - f(x, y)$$

respectively.

The divergence of a vector field F is approximated by

$$\nabla \cdot F \approx \big(F_x(x, y) - F_x(x-1, y)\big) + \big(F_y(x, y) - F_y(x, y-1)\big)$$

Therefore, $-\nabla J(f)$ is discretized as

$$-\nabla J(f) = -(f-g) + \lambda \nabla \cdot \left(\frac{\nabla f}{|\nabla f|}\right)$$

$$\approx -(\mathrm{f}-\mathrm{g}) + \lambda\left(\left(\frac{\nabla f}{|\nabla f|}\right)_x (x,y) - \left(\frac{\nabla f}{|\nabla f|}\right)_x (x-1,y)\right)$$

$$+\left(\left(\frac{\nabla f}{|\nabla f|}\right)_y (x,y) - \left(\frac{\nabla f}{|\nabla f|}\right)_y (x,y-1)\right)$$

$$\approx -\big(f(x,y)-g(x,y)\big)$$

$$+\lambda \frac{f(x+1,y) - f(x,y)}{\sqrt{\big(f(x+1,y) - f(x,y)\big)^2 + \big(f(x,y+1) - f(x,y)\big)^2}}$$

$$-\lambda \frac{f(x,y) - f(x-1,y)}{\sqrt{\big(f(x,y) - f(x-1,y)\big)^2 + \big(f(x-1,y+1) - f(x-1,y)\big)^2}}$$

$$+\lambda \frac{f(x,y+1) - f(x,y)}{\sqrt{\big(f(x+1,y) - f(x,y)\big)^2 + \big(f(x,y+1) - f(x,y)\big)^2}}$$

$$-\lambda \frac{f(x,y) - f(x,y-1)}{\sqrt{\big(f(x+1,y-1) - f(x,y-1)\big)^2 + \big(f(x,y) - f(x,y-1)\big)^2}}$$

Step 3:

Repeat step 2 for a fixed number of times. (In all the programs that we ran, we set the number to be 2500 for simplicity)

Also, we plotted the energy of the image at each iteration to see if the solution has converged yet. We found that 2500 iterations are more than enough for all the models stated in this summary.

Below is a part of the code of the TV model for updating the gradient,

```
grad = -(f - img);
% assume the image extend periodically. Note that in here, x is the
% horizontal direction (columns) and y is the verticle direction (rows)
f_forward_x = circshift(f, 1, 2);
f_forward_y = circshift(f, 1, 1);
f_backward_x = circshift(f, -1, 2);
f_backward_y = circshift(f, -1, 1);

% differences
forward_diff_x = f_forward_x - f;
forward_diff_y = f_forward_y - f;
backward_diff_x = f_backward_x - f;
backward_diff_y = f_backward_y - f;

% for the gradient magnatude
% backward x, forward y
a = circshift(f, [1, -1]) - f_backward_x;
% forward x, backward y
b = circshift(f, [-1, 1]) - f_backward_y;
grad = grad + lambda * (forward_diff_x) ./ sqrt(epsilon + forward_diff_x .^2 + forward_diff_y .^2);
grad = grad + lambda * (backward_diff_x) ./ sqrt(epsilon + backward_diff_x .^2 +  a.^2);
grad = grad + lambda * (forward_diff_y) ./ sqrt(epsilon + forward_diff_x .^2 + forward_diff_y .^2);
grad = grad + lambda * (backward_diff_y) ./ sqrt(epsilon + backward_diff_y .^2 +  b.^2);
```

7. <u>Proposed model 1:</u>

From part 5, we see that the modified TV model can only improve the detail-preserving ability of the TV model but cannot handle the staircase effect. Therefore, starting from this part, we propose 2 models that aim to reduce the staircase effect of the TV model.

We modified the TV model as follows,

$$J(f) = \frac{1}{2} \int\int_{\Omega} \big(f(x,y) - g(x,y)\big)^2 \, dx\, dy + \beta \int\int_{\Omega} |\nabla^2 f|(x,y) dx\, dy$$

, where $\beta > 0$ is a parameter.

Now, define

$$L\big(x, y, f, f_x, f_y, f_{xy}, f_{xx}, f_{yy}\big) = \frac{1}{2}\big(f(x,y) - g(x,y)\big)^2 + \beta|\nabla^2 f|(x,y)$$

Then,

$$J(f) = \int\int_{\Omega} L\big(x, y, f, f_x, f_y, f_{xy}, f_{xx}, f_{yy}\big)\, dx\, dy$$

Our motivation of proposing this model is that, from the introduction part, we see that the TV model produces staircase effect because the gradient of the optimal solution is zero (or nearly zero) at many points. Therefore, we try to measure the oscillation of the pixel values using the Laplacian magnitude instead of the gradient magnitude, in order words, we replace the regularization from $||\nabla f||_1$ to $||\nabla^2 f||_1$, that is the Laplacian of f. By this modification, we can prevent the magnitude gradient of the image to be zero at many pixels and we hope this could reduce the staircase effect.

Numerical method used

Same as the TV model, we apply the gradient descent method to solve the minimization problem. Step 1 and Step 3 are the same. Step 2 is almost the same except for the gradient. Again, the gradient for this problem can be found by finding the Euler-Lagrange equation of J(f). Since $L(x, y, f, f_x, f_y, f_{xy}, f_{xx}, f_{yy})$ depends on a two-dimensional function and its second-order derivatives, the Euler-Lagrange equation of J(f) is given by

$$\frac{\partial L}{\partial f} - \frac{\partial}{\partial x}\frac{\partial L}{\partial f_x} - \frac{\partial}{\partial y}\frac{\partial L}{\partial f_y} + \frac{\partial^2}{\partial x \partial y}\frac{\partial^2 L}{\partial f_x \partial f_y} + \frac{\partial^2}{\partial x^2}\frac{\partial L}{\partial f_{xx}} + \frac{\partial^2}{\partial y^2}\frac{\partial L}{\partial f_{yy}} = 0$$

Since L does not depend on f_x and f_y, we can simplify the above equation as

$$\frac{\partial L}{\partial f} + \frac{\partial^2}{\partial x^2}\frac{\partial L}{\partial f_{xx}} + \frac{\partial^2}{\partial y^2}\frac{\partial L}{\partial f_{yy}} = 0$$

Note that

$$\frac{\partial L}{\partial f} = f - g$$

$$\frac{\partial L}{\partial f_{xx}} = \beta \frac{f_{xx}}{|f_{xx} + f_{yy}|}$$

And

$$\frac{\partial L}{\partial f_{yy}} = \beta \frac{f_{yy}}{|f_{xx} + f_{yy}|}$$

Therefore,

$$\frac{\partial L}{\partial f} + \frac{\partial^2}{\partial x^2}\frac{\partial L}{\partial f_{xx}} + \frac{\partial^2}{\partial y^2}\frac{\partial L}{\partial f_{yy}}$$

$$= (f - g) + \beta \nabla^2 \left(\frac{\nabla^2 L}{|\nabla^2 L|}\right)$$

Hence,

$$-\nabla J(f) = -EulerLagrangeEquation$$
$$= -(f-g) - \beta\nabla^2\left(\frac{\nabla^2 L}{|\nabla^2 L|}\right)$$

Using the same approximation scheme as in the previous part, the Laplacian can be approximated by

$$\nabla^2 f \approx -4\mathrm{f}(\mathrm{x},\mathrm{y}) + \mathrm{f}(\mathrm{x}-1,\mathrm{y}) + \mathrm{f}(\mathrm{x}+1,\mathrm{y}) + \mathrm{f}(\mathrm{x},\mathrm{y}-1) + \mathrm{f}(\mathrm{x},\mathrm{y}+1)$$

Hence, the discretization of $-\nabla J(f)$ is given by

$$-\nabla J(f) = -(f-g) - \beta\nabla^2\left(\frac{\nabla^2 L}{|\nabla^2 L|}\right)$$

$$\approx -(f-g) - \beta\Bigg(-4\nabla^2\left(\frac{\nabla^2 L}{|\nabla^2 L|}\right)(x,y) + \nabla^2\left(\frac{\nabla^2 L}{|\nabla^2 L|}\right)(x-1,y)$$
$$+ \nabla^2\left(\frac{\nabla^2 L}{|\nabla^2 L|}\right)(x+1,y) + \nabla^2\left(\frac{\nabla^2 L}{|\nabla^2 L|}\right)(x,y-1)$$
$$+ \nabla^2\left(\frac{\nabla^2 L}{|\nabla^2 L|}\right)(x,y+1)\Bigg)$$

$$\approx -\big(f(x,y) - g(x,y)\big)$$
$$+ 4\beta\frac{\big(f(x+1,\ y) + f(x-1,y) + f(x,y+1) + f(x,y-1) - 4f(x,y)\big)}{|f(x+1,\ y) + f(x-1,y) + f(x,y+1) + f(x,y-1) - 4f(x,y)|}$$
$$- \ \beta\frac{\big(f(x,\ y) + f(x-2,y) + f(x-1,y+1) + f(x-1,y-1) - 4f(x-1,y)\big)}{|f(x,\ y) + f(x-2,y) + f(x-1,y+1) + f(x-1,y-1) - 4f(x-1,y)|}$$
$$- \ \beta\frac{\big(f(x+2,\ y) + f(x,y) + f(x+1,y+1) + f(x+1,y-1) - 4f(x+1,y)\big)}{|f(x+2,\ y) + f(x,y) + f(x+1,y+1) + f(x+1,y-1) - 4f(x+1,y)|}$$
$$- \ \beta\frac{\big(f(x+1,\ y-1) + f(x-1,y-1) + f(x,y) + f(x,y-2) - 4f(x,y-1)\big)}{|f(x+1,\ y-1) + f(x-1,y-1) + f(x,y) + f(x,y-2) - 4f(x,y-1)|}$$
$$- \ \beta\frac{(f(x+1,\ y+1) + f(x-1,y+1) + f(x,y+2) + f(x,y) - 4f(x,y+1))}{|(f(x+1,\ y+1) + f(x-1,y+1) + f(x,y+2) + f(x,y) - 4f(x,y+1))|}$$

Sample Code

Below shows the code that computes the last 5 terms in the above formula.

```
if strcmp(mode, 'laplacian')
    % f(x+2, y), f(x, y+2), f(x, y+2), f(x, y-2)
    f_forward_x2 = circshift(f, 2, 2);
    f_forward_y2 = circshift(f, 2, 1);
    f_backward_x2 = circshift(f, -2, 2);
    f_backward_y2 = circshift(f, -2, 1);
    % f(x+1, y+1), f(x-1, y-1), f(x+1, y-1), f(x-1, y+1)
    f_forwx_forwy = circshift(f, [1 1]);
    f_backx_backy = circshift(f, [-1 -1]);
    f_forwx_backy = circshift(f, [-1 1]);
    f_backx_forwy = circshift(f, [1 -1]);

    % laplacians evaluated at points (x,y) (x-1,y), (x+1,y), (x,y-1), (x,y+1)
    laplacian_x_y = f_forward_x + f_backward_x + f_forward_y + f_backward_y - 4 * f;
    laplacian_backx_y = f + f_backward_x2 + f_backx_forwy + f_backx_backy - 4 * f_backward_x;
    laplacian_forwx_y = f_forward_x2 + f + f_forwx_forwy + f_forwx_backy - 4 * f_forward_x;
    laplacian_x_backy = f_forwx_backy + f_backx_backy + f + f_backward_y2 - 4 * f_backward_y;
    laplacian_x_forwy = f_forwx_forwy + f_backx_forwy + f_forward_y2 + f - 4 * f_forward_y;

    grad = grad + beta * 4 * (laplacian_x_y) ./ (epsilon + abs(laplacian_x_y));
    grad = grad - beta * (laplacian_backx_y) ./ (epsilon + abs(laplacian_backx_y));
    grad = grad - beta * (laplacian_forwx_y) ./ (epsilon + abs(laplacian_forwx_y));
    grad = grad - beta * (laplacian_x_backy) ./ (epsilon + abs(laplacian_x_backy));
    grad = grad - beta * (laplacian_x_forwy) ./ (epsilon + abs(laplacian_x_forwy));
end
```

Results of proposed model 1

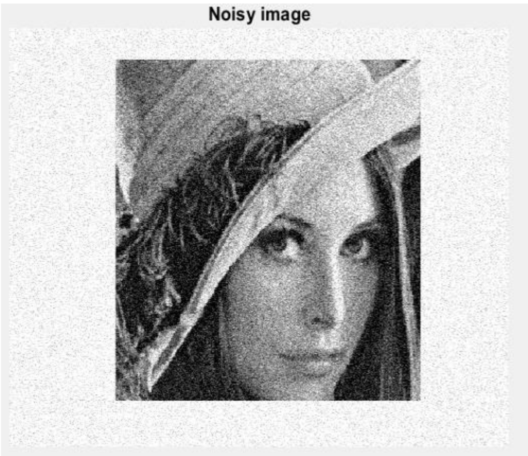


Fig 1.1

Fig 1.2

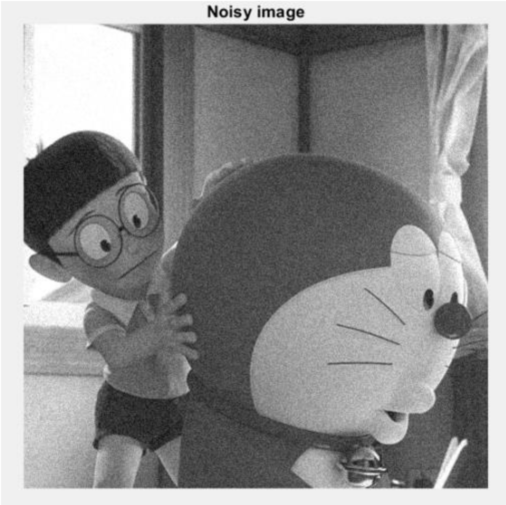


Fig 1.3

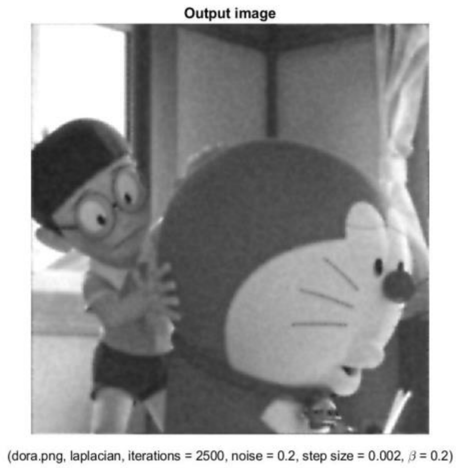


Fig 1.4

Fig 1.5

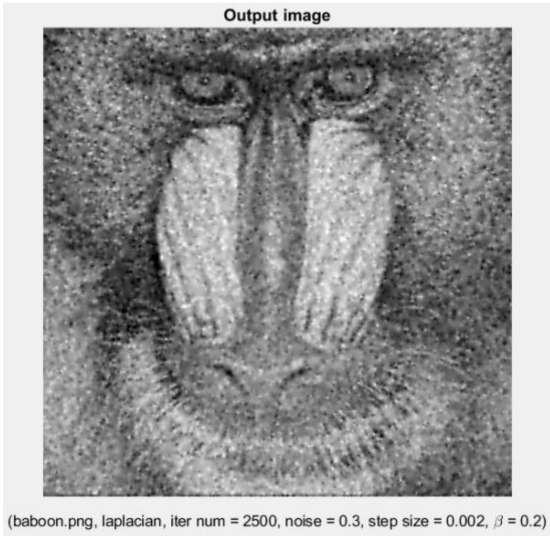
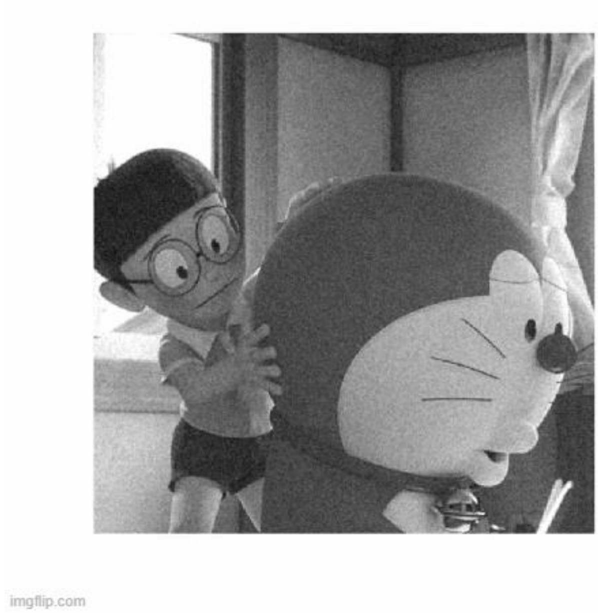


Fig 1.6

From Fig 1.1 to Fig 1.6, we see that the proposed model can really denoise images. Although there is still some noise remaining in the output images, we can see that there is no staircase effect. The results above show that the motivation of the first proposed model makes sense, and the first proposed model can really remove some denoise without creating staircase effect.

Role of the parameter β

Gif 1.1



Gif 1.2

From experiments, it is observed that when β is small (less than 1), then the proposed model can denoise an noisy image. However, when β becomes large, the noise becomes even stronger, and the image will eventually be destroyed.

Now, we can summarize the performance of the proposed model 1.

Advantages:

- Noise is reduced
- No staircase effect

Disadvantages:

- The denoising ability is not as good as the TV model

- The parameter β has to be chosen carefully

8. Proposed model 2:

Since the denoising ability of the first proposed model is not as strong as the denoising ability of the TV model but it does not create any staircase effect, this motivates us to propose the second model. We further modified the above model by combining it to the TV model.

$$J(f) = \frac{1}{2} \int\int_{\Omega} \left(f(x,y) - g(x,y)\right)^2 dx\, dy + \lambda \int\int_{\Omega} |\nabla f|(x,y) dx\, dy + \beta \int\int_{\Omega} |\nabla^2 f|(x,y) dx\, dy$$

, where $\lambda > 0$ and $\beta > 0$ are the parameters

The motivation of proposing this model is that we try to combine the advantages of the TV model and the proposed model 1.

Numerical method used

Again, gradient descent method is used for the proposed model 2. We follow step 1 to step 3 stated above. The only difference is the gradient ∇J.
However, since gradient operator is linear and the above model is just the combination of the TV model and the proposed model 1, the gradient of this model can be found easily using previous results on the gradient of the TV model and the gradient of the proposed model 1.

Sample Code

Below is a portion of our code for updating the gradient. As the gradient operator is linear, we can reuse the code in the proposed model 1 and the TV model.

```
grad = grad + beta * 4 * (laplacian_x_y) ./ (epsilon + abs(laplacian_x_y));
grad = grad - beta * (laplacian_backx_y) ./ (epsilon + abs(laplacian_backx_y));
grad = grad - beta * (laplacian_forwx_y) ./ (epsilon + abs(laplacian_forwx_y));
grad = grad - beta * (laplacian_x_backy) ./ (epsilon + abs(laplacian_x_backy));
grad = grad - beta * (laplacian_x_forwy) ./ (epsilon + abs(laplacian_x_forwy));

grad = grad + lambda * (((forward_diff_x) ./ sqrt(epsilon + forward_diff_x .^2 + forward_diff_y .^2)) + 4 * forward_diff_x .* (forward_diff_x.^2+forward_diff_y.^2));
grad = grad + lambda * (((backward_diff_x) ./ sqrt(epsilon + backward_diff_x .^2 +  a.^2)) + 4 * backward_diff_x .* (backward_diff_x.^2+a.^2));
grad = grad + lambda * (((forward_diff_y) ./ sqrt(epsilon + forward_diff_x .^2 + forward_diff_y .^2)) + 4 * forward_diff_y .* (forward_diff_x.^2+forward_diff_y.^2));
grad = grad + lambda * (((backward_diff_y) ./ sqrt(epsilon + backward_diff_y .^2 +  b.^2)) + 4 * backward_diff_y .* (backward_diff_y.^2+b.^2));
```

Results of proposed model 2

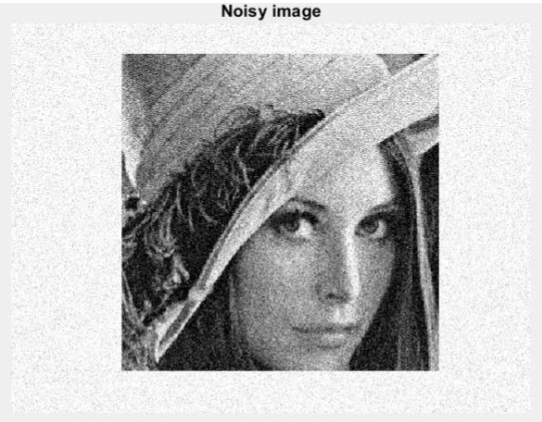


Fig 2.1

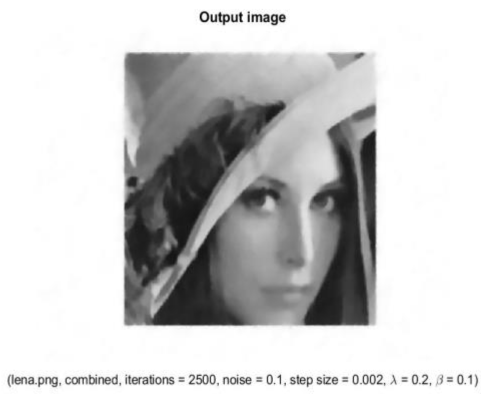


Fig 2.2

Fig 2.3

Fig 2.4

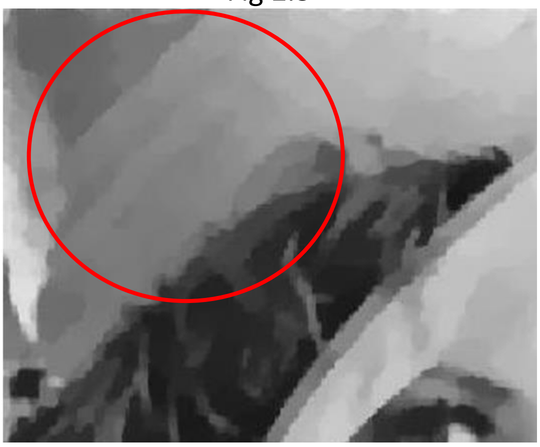


Fig 2.5

Fig 2.6

From Fig 2.1 and Fig 2.2, we can see that the proposed model has better denoising ability compare to the proposed model 1. Fig 2.3 and Fig 2.5 are the results from TV model using the same λ as in Fig 2.2, that is $\lambda = 0.2$. Fig 2.4 and Fig 2.6 are the zoomed images of Fig 2.2. We can clearly see that the staircase effect in Fig 2.4 and Fig 2.6 is less serious than that in Fig 2.3 and Fig 2.5. The pixel colors transit smoothly in Fig 2.4 and Fig 2.6. This example shows that the proposed model 2 successfully combine the advantages of both the TV model and the proposed model 1.

Below are some more examples.

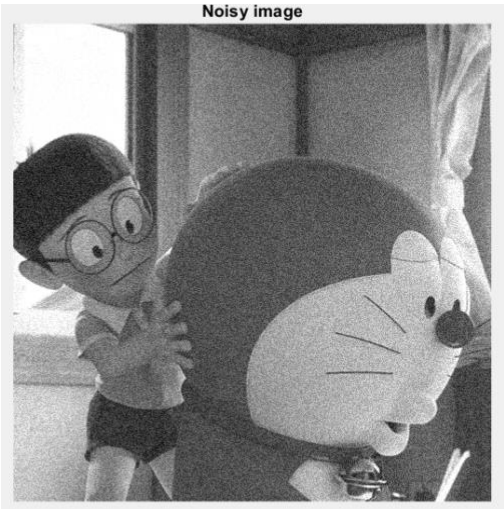


Fig 2.7

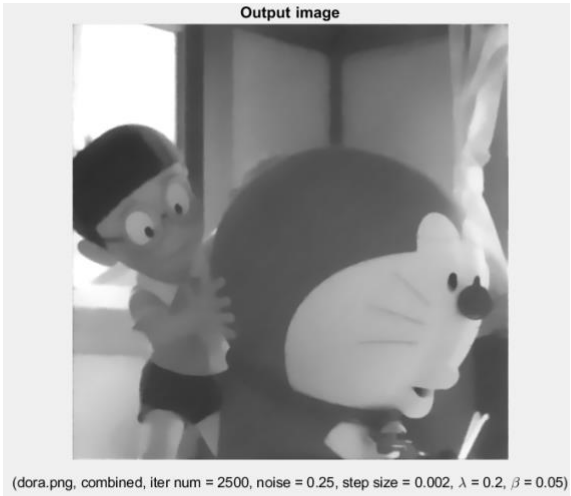


Fig 2.8

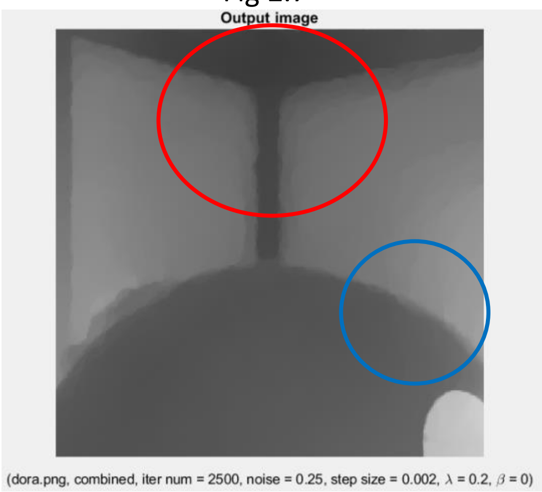


Fig 2.9

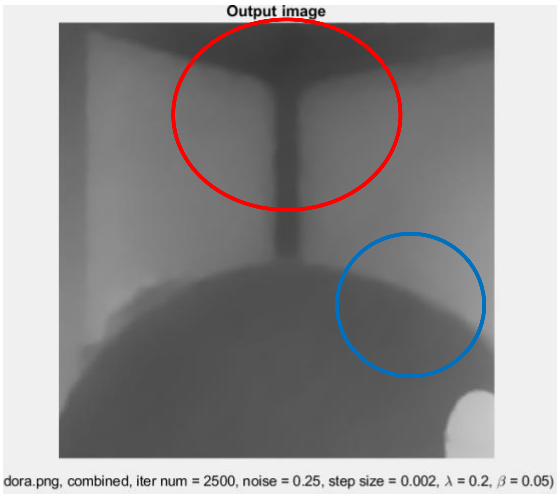


Fig 2.10

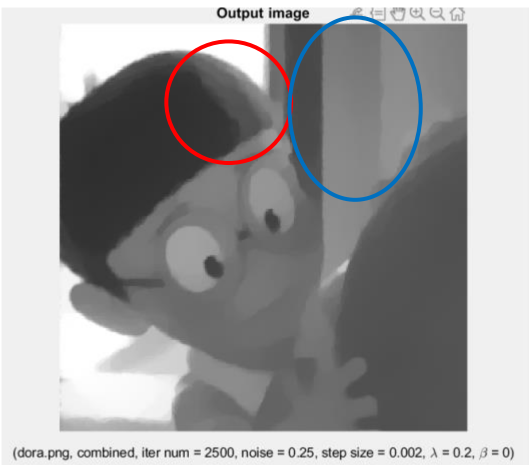


Fig 2.11

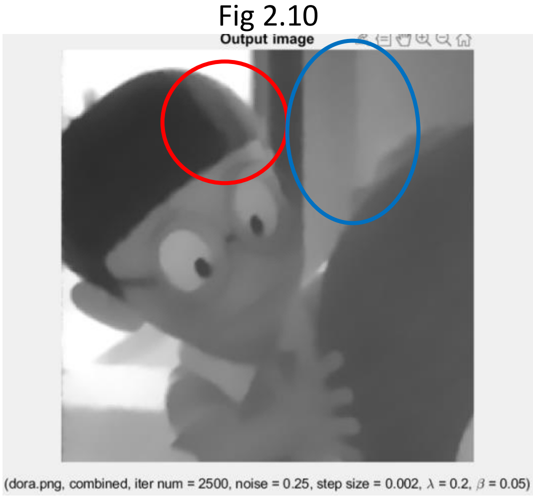


Fig 2.12

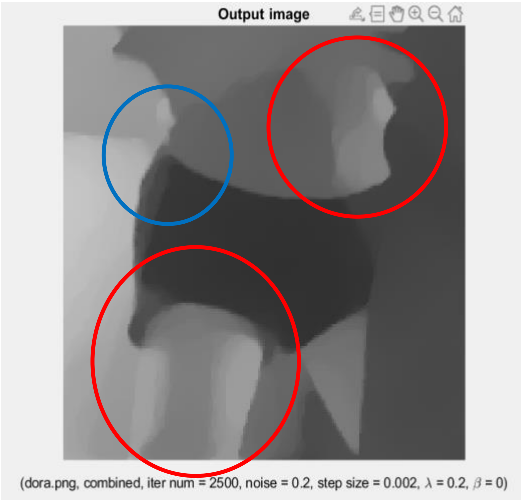


Fig 2.13

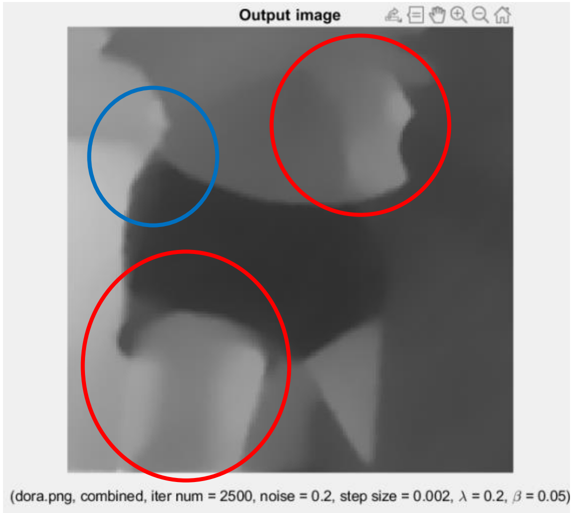


Fig 2.14

Similar to the previous example, Fig 2.9, 2.11 and 2.13 are the results of TV model using the same λ as in Fig 2.8. Fig 2.10, 2.12 and 2.14 are the zoomed images of Fig 2.8. Again, we see that the denoising ability of the proposed model 2 is similar to that of the TV model under the same λ, which is set to be 0.2 here. However, the staircase effect becomes less obvious in the proposed model 2, where the parameter β is only set to be 0.05. (Please compare the red circles)

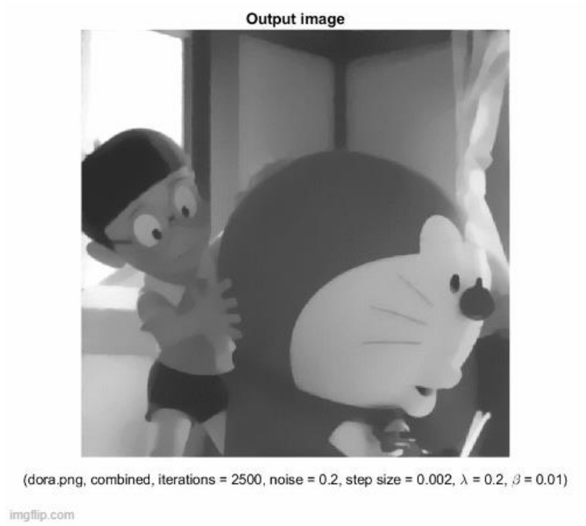
Also, this example also shows that the proposed model 2 can keep the sharp edges well. The proposed model 2 can smooth out the "fake edges" caused by the staircase effect without smoothing the true edges. (See the blue circles)

<u>Role of the parameters (λ and β)</u>

The role of λ here is the same as that in the TV model, that is, it dominates the denoising ability of the model. The larger λ is, the smoother the output image will be. In Gif 2.3 and 2.4, the outputs at each λ are shown, where the parameter β is fixed to be 0.1.

The role of β here is also the same as that in the first proposed model. If β is small (usually less than 1 in our experiments), then the last term of the model can remove some noise in the image without causing the staircase effect. However, if β is too large, then the noise becomes even stronger, and the image will eventually be corrupted completely. In Gif 2.1 and 2.2 the outputs at each β are shown, where the parameter λ is fixed to be 0.1. Fig 2.15 and 2.16 are the outputs when β is 100, these 2 images illustrate that β cannot set to be too large, otherwise, the output images will be corrupted.

Gif 2.1



Gif 2.2

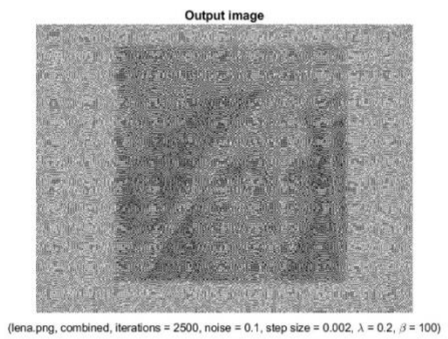


Fig 2.15

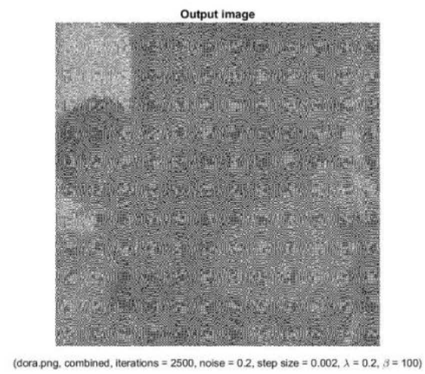
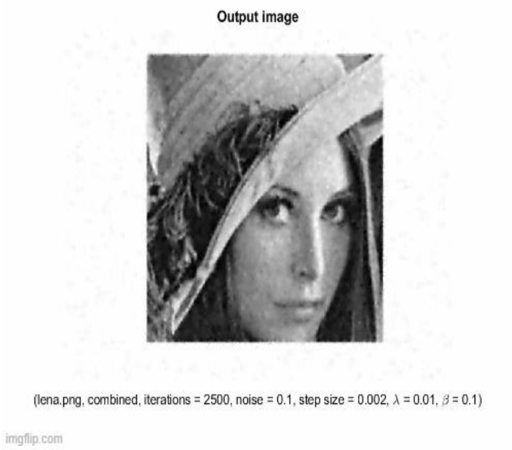
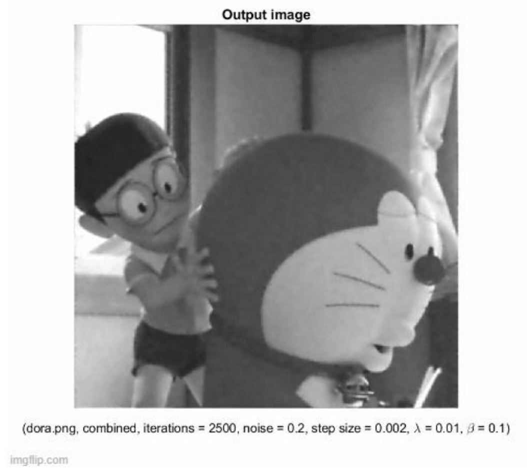


Fig 2.16



Gif 2.3



Gif 2.4

Now, we can summarize the performance of the proposed model 2.

Advantages

- Better denoising effect compared to the proposed model 1
- The staircase effect is not as obvious as that of the TV model (under the same λ)

Disadvantage

- The parameter β has to be chosen carefully.

9. Conclusion

In this project, our group studied the staircase effect of the TV model and an existing modification of the TV model. Then, I proposed 2 models that aim to reduce noise without staircase effect. The proposed model 1 can denoise without staircase effect but the results are not satisfactory. Therefore, I tried to combine it with the TV model, hoping to combine the advantages of the 2 models. The performance of the proposed model 2 is satisfactory, it has better denoising ability than the proposed model 1 and has less staircase effect than the TV model.

Reference

Aubert, Gilles, and Pierre Kornprobst. Mathematical Problems in Image Processing. Second ed. Vol. 147. New York, NY: Springer New York. Applied Mathematical Sciences. Web.

Rudin, Leonid I, Stanley Osher, and Emad Fatemi. "Nonlinear Total Variation Based Noise Removal Algorithms." Physica. D 60.1 (1992): 259-68. Web.

You, Y.-L, and M. Kaveh. "Fourth-order Partial Differential Equations for Noise Removal." IEEE Transactions on Image Processing 9.10 (2000): 1723-730. Web.