# Effect of Various Denoisers in Different Applications of Plug-and-Play ADMM

**Marvin Jesse, AAE, MS** [1]

## Abstract

The Plug-and-Play ADMM (PnP) ADMM algorithm is a powerful image restoration algorithm, which works by integrating advanced denoisers to be incorporated into the physical forward models. Even though it has been sucessfully used in numerous applications, a question arises: does a better denoiser translate to a better PnP ADMM solution? There has been little efforts in answering this question. Therefore, the goal of this paper is to try to answer that question. By implementing various denoisers in PnP ADMM for some applications, we will see that the best denoiser does not provide the best PnP ADMM solution. Numerous experiments in different settings will be conducted to verify this hypothesis

## 1. Introduction

### 1.1. ADMM Review

The Plug-and-Play (PnP) ADMM is a subset of the alternating direction method of multiplier (ADMM) algorithms, which has been extensively used in many applications, such as image restoration (Venkatakrishnan et al., 2013a) and signal recovery problems (Dar et al., 2016). In general, the ADMM algorithm is trying to break an optimization problem into some sub-problems. Consider the following optimization problem:

$$\hat{x} = \arg\min_x f(x) + g(x), \tag{1}$$

where $f(x)$ and $g(x)$ are arbitrary functions of a variable $x$. The ADMM breaks problem 1 into 3 sub-problems:

$$x^{(k+1)} = \arg\min_x f(x) + \frac{\rho}{2}||x - \tilde{x}||^2, \tag{2}$$

$$v^{(k+1)} = \arg\min_v g(v) + \frac{\rho}{2}||v - \tilde{v}||^2, \tag{3}$$

$$\overline{u}^{(k+1)} = \overline{u}^{(k)} + \left(x^{(k+1)} - v^{(k+1)}\right), \tag{4}$$

where $v^{(k)}$ is a dual variable, $\overline{u}^{(k)} = \frac{1}{\rho}u^{(k)}$ is a scaled Lagrange multiplier, $\tilde{x}^{(k)} = v^{(k)} - \overline{u}^{(k)}$, $\tilde{v}^{(k)} = x^{(k+1)} + \overline{u}^{(k)}$, and $\rho$ is some constant.

### 1.2. Problem Formulation

In order to solve an image restoration problem using PnP ADMM, the problem 3 can be interpreted as a denoising problem, and it can be replaced by any widely used denoisers. Therefore, it is clear that different denoisers will result in different Pnp ADMM solutions. A question arising from this statement is that "Is it possible to guarantee a better PnP ADMM solution using a better denoiser?". First, we need to define what it means to have a better denoiser. A denoiser is better than other denoiser if the mean squared error on testing is uniformly lower at each testing noise level (Guo, 2022). It means that a better denoiser corresponds to a higher peak signal-to-noise ratio (PSNR) value. Next, to answer the aforementioned question, we would like to develop a comparison scheme among different denoisers for some standard inverse problems. In this paper, we will use well-known denoisers to solve some PnP ADMM problems, namely image deblurring, image super resolution, and image inpainting, and try to deduce whether a better denoiser translates to a better PnP ADMM solution in these applications. Our intuition is that the answer to the question is yes, but we will prove by in-depth experiments that this conjecture is wrong.

## 2. Related Work

### 2.1. Problem Description

In this subsection, we would like to formulate the image deblurring, inpainting, and super resolution as optimization problems:

#### 2.1.1. Image deblurring

Image deblurring is a classic problem in the computer vision community, which aims to recover a sharp image from a blurred image. The blur can be caused by many factors, such as lack of focus, camera shake, and fast target motion (Zhang et al., 2022). This problem can be described mathematically as the following:

$$\hat{x} = \arg\min_x \frac{1}{2}||Hx - y||^2 + \lambda g(x), \tag{5}$$

where $x$ is the original image, $y$ is the observed image, and $H \in \mathbb{R}^{n\times n}$ is a circulant matrix denoting the blur.

#### 2.1.2. IMAGE INPAINTING

Shared images in social network might contain unwanted objects, such as signature or emoticons. Removing these objects is a well-known problem in computer vision research, and one way to tackle this problem is by image inpainting. Previously, image inpainting was applied on old images in order to remove small scratches. Now, it is used for removing artifact objects that can be added to the images (Elharrouss et al., 2020). The problem can be formulated as:

$$\hat{x} = \arg\min_x \frac{1}{2}||Sx - y||^2 + \lambda g(x), \tag{6}$$

where $x$ is the original image, $y$ is the observed image, and $S \in \mathbb{R}^{n\times n}$ is a diagonal masking matrix. $S_{ii} = 1$ if the $i$-th sampled is selected, and $S_{ii} = 0$ otherwise.

#### 2.1.3. IMAGE SUPER RESOLUTION

Super-resolution is the process of generating a raster image with a higher resolution than its source, where the source can consist of one or more images or frames (van Ouwerkerk, 2006). The problem is described as follows:

$$\hat{x} = \arg\min_x \frac{1}{2}||SHx - y||^2 + \lambda g(x), \tag{7}$$

where $x$ is the original image, $y$ is the observed image, $S \in \mathbb{R}^{m\times n}$ is a binary matrix denoting the $K$-fold downsampling, where $m < n$, and $H \in \mathbb{R}^{n\times n}$ is a convolution matrix representing the anti-aliasing filter.

### 2.2. PnP ADMM Overview

As described in Section 2.1, all of the problems (5), (6), and (7) is the same as problem (1). Therefore, one way to solve them is to use PnP ADMM. The PnP ADMM algorithm can be thought as solving a standard inverse problem: we try to find a clean, latent original image $x \in \mathbb{R}^n$, given the observed image $y \in \mathbb{R}^n$ where it is corrupted by some linear forward model and noise. Here, the solution $\hat{x} \in \mathbb{R}^n$ should "best explain" the observation $y$. This problem can be considered as a maximum-a-posteriori (MAP) estimation in the following:

$$\begin{aligned}\hat{x} &= \arg\max_x p(x|y) \\ &= \arg\min_x -log\ p(y|x) - log\ p(x),\end{aligned} \tag{8}$$

where $p(x|y)$ is a conditional probability defining the forward imaging model, and $p(x)$ is the probability distribution of the latent image. It is clear that the MAP problem in (8) is equivalent to this optimization problem:

$$\hat{x} = \arg\min_x f(x) + \lambda g(x) \tag{9}$$

by setting $f(x) = -log\ p(y|x)$ and $g(x) = (-1/\lambda) log\ p(x)$, and $\lambda$ is some regularization parameter. Here, (9) is a generic unconstrained optimization problem, thus enabling us to use ADMM. The general idea of ADMM is to break the optimization problem into smaller pieces, each of which are easier to handle. Specifically, in problem (9), we introduce a new variable $v$ such that we can convert the unconstrained optimization problem into a constrained one in the following fashion:

$$(\hat{x}, \hat{v}) = \arg\min_{x.v} f(x) + \lambda g(v), \text{ subject to, } x = v. \tag{10}$$

As in the method of multiplier, we can consider the following augmented Lagrangian function:

$$\mathcal{L}(x, v, u) = f(x) + \lambda g(v) + u^T(x - v) + \frac{\rho}{2}||x - v||^2, \tag{11}$$

where $u$ is the Lagrange multiplier, and $\rho$ is a positive constant. The solution to (10) is the saddle point of $\mathcal{L}$, which can be found by solving three subproblems:

$$x^{(k+1)} = \arg\min_x f(x) + \frac{\rho}{2}||x - \tilde{x}^{(k)}||^2, \tag{12}$$

$$v^{(k+1)} = \arg\min_v \lambda g(v) + \frac{\rho}{2}||v - \tilde{v}^{(k)}||^2, \tag{13}$$

$$\overline{u}^{(k+1)} = \overline{u}^{(k)} + \left(x^{(k+1)} - v^{(k+1)}\right), \tag{14}$$

where $\overline{u}^{(k)} = (1/\rho)u^{(k)}$ is the scaled Lagrange multiplier, $\tilde{x}^{(k)} = v^{(k)} - \overline{u}^{(k)}$, and $\tilde{v}^{(k)} = x^{(k+1)} + \overline{u}^{(k)}$. As clearly shown in (12) and (13), the value of $x$ and $v$ are updated in an alternating fashion, explaining the term *alternating direction*. Under some mild assumptions, it is shown that the solution from the iteration defined in (12), (13), and (14) converges to the true solution defined in (10) (Boyd et al., 2011).

As clearly mentioned in (Chan et al., 2016), (12) can be regarded as an inversion step since it involves the forward imaging model $f(x)$, while (13) is a denoising step as it involves prior $g(v)$. In order to verify the latter, let us define $\sigma = \sqrt{\lambda/\rho}$, and it is straightforward to show that (13) is equivalent to

$$v^{(k+1)} = \arg\min_v g(v) + \frac{1}{2\sigma^2}||v - \tilde{v}^{(k)}||^2. \tag{15}$$

Here, the optimization problem (15) tries to minimize the "residual" image, i.e., the difference between the clean image $v$ and the noisy image $\tilde{v}^{(k)}$, using prior $g(v)$.

According to (Romano et al., 2017), the denoising operator is defined as the solution to the following optimization problem, i.e,

$$\begin{aligned}D_\sigma(\tilde{z}) &= \arg\min_z \frac{1}{2\sigma^2}||\tilde{z} - z||^2 + \kappa p(z), \\ \tilde{z} &= z + w,\end{aligned} \tag{16}$$

where $D_\sigma(z)$ is the denoising operator, $\tilde{z}$ is the noisy image, $z$ is the denoised image, $w$ is additive white Gaussian noise, and $\sigma$ specifies the denoising strength. Comparing (15) and (18), it can be inferred that (13) is a denoising operator that operates on $\tilde{v}^{(k)}$, i.e.,

$$v^{(k+1)} = D_\sigma(\tilde{v}^{(k)}). \tag{17}$$

Since we can use different denoisers for this purpose (this can be done by replacing the denoising operator defined in (17) by the appropriate denoiser), the resulting algorithm is then called Plug-and-Play ADMM.

### 2.3. PnP ADMM Algorithm Description

The PnP ADMM algorithm is summarized as follows:

$$x^{(k+1)} = \arg\min_x f(x) + \frac{\rho_k}{2}||x - (v^{(k)} - u^{(k)})||^2, \tag{18}$$

$$v^{(k+1)} = D_{\sigma_k}(x^{(k+1)} + u^{(k)}), \tag{19}$$

$$u^{(k+1)} = u^{(k)} + (x^{(k+1)} - v^{(k+1)}), \tag{20}$$

where $D_{\sigma_k}$ is a denoising operator(or called "denoiser" for short), and $\sigma_k = \sqrt{\lambda/\rho_k}$ is a parameter controlling the denoiser's strength. Here, $\lambda$ can be considered similar to a regularization parameter in conventional ADMM algorithm.

Here, $\rho_k$ determines the optimization cost in (18), so it is important to characterize how $\rho_k$ changes the solution of the PnP ADMM. In (Venkatakrishnan et al., 2013b), the time-varying parameter $\rho_k$ is set to be a constant, i.e., $\rho_k = \rho$ for some $\rho$. However, this rule does not work when the denoiser $D_{\sigma_k}$ is non-expansive and has symmetric gradient (Chan et al., 2016). Therefore, another rule is proposed called *monotone update rule*, which is defined as

$$\rho_{k+1} = \gamma\rho_k, \ \forall k, \tag{21}$$

where $\gamma > 1$ is a constant. Another option is to use an *adaptive update rule* by considering the residue $\Delta_k$ as

$$\Delta_{k+1} = \frac{1}{\sqrt{n}}\Big(||x^{(k+1)} - x^{(k)}||_2 + ||v^{(k+1)} - v^{(k)}||_2 + ||u^{(k+1)} - u^{(k)}||_2\Big). \tag{22}$$

Then, we can formulate the update rule as

$$\rho_{k+1} = \begin{cases} \gamma\rho_k, & \text{if } \Delta_{k+1} \geq \eta\Delta_k \\ \rho_k, & \text{otherwise} \end{cases} \tag{23}$$

where $\eta \in [0, 1)$ is a constant. It is shown that the PnP ADMM solution resulted from the adaptive update rule is better (Chan et al., 2016).

In terms of the fixed point convergence, this algorithm will converge, i.e., $\Delta_k \to 0$ as $k \to \infty$. Therefore, we can use the residual term as a stopping criterion, i.e., we choose to terminate the iteration when

$$\Delta_{k+1} \leq tol, \tag{24}$$

where $tol$ is some tolerance level. Another alternative for the stopping condition is

$$\max\{\epsilon_1, \epsilon_2, \epsilon_3\} \leq tol/3, \tag{25}$$

where $\epsilon_1 = ||x^{(k+1)} - x^{(k)}||_2/\sqrt{n}$, $\epsilon_2 = ||v^{(k+1)} - v^{(k)}||_2/\sqrt{n}$, and $\epsilon_3 = ||u^{(k+1)} - u^{(k)}||_2/\sqrt{n}$. In general, $tol = 10^{-3}$ is sufficient for the purpose of image restoration (Chan et al., 2016). The more important parameters that influence our results are the update ratio $\gamma$ and the initial value $\rho_0$. When $\gamma \in (1, 2)$ and $\rho_0 \in (10^{-5}, 10^{-2})$, it is acceptable to set $tol = 10^{-3}$. An additional stopping criterion that we usually add is that we set the number of maximum iteration, i.e.,

$$k > maxIter, \tag{26}$$

where $maxIter$ is the maximum number of iterations. It is desirable in case that we need to terminate our iteration early. The overall algorithm of the PnP ADMM is summarized in Algorithm 1.

**Algorithm 1** Plug-and-Play ADMM

**Require:** $\rho_0, \gamma, maxIter, tol, \eta$
  **while** $k \leq maxIter$ and $\Delta_k > tol$ **do**
    $x^{(k+1)} = \arg\min_x f(x) + \frac{\rho_k}{2}||x - (v^{(k)} - u^{(k)})||^2$
    $v^{(k+1)} = D_{\sigma_k}(x^{(k+1)} + u^{(k)})$
    $u^{(k+1)} = u^{(k)} + (x^{(k+1)} - v^{(k+1)})$
    $\epsilon_1 = ||x^{(k+1)} - x^{(k)}||_2/\sqrt{n}$
    $\epsilon_2 = ||v^{(k+1)} - v^{(k)}||_2/\sqrt{n}$
    $\epsilon_3 = ||u^{(k+1)} - u^{(k)}||_2/\sqrt{n}$
    $\Delta_{k+1} = \epsilon_1 + \epsilon_2 + \epsilon_3$
    **if** $\Delta_{k+1} \geq \eta\Delta_k$ **then**
      $\rho_{k+1} = \gamma\rho_k$
    **else**
      $\rho_{k+1} = \rho_k$
    **end if**
  **end while**

## 3. Method

### 3.1. Experimental Framework

In order to verify our hypothesis, which is whether a better denoisers implies a better solution to PnP ADMM, we conduct experiments on the performance of the PnP ADMM solutions among different denoisers in three applications: image deblurring, image inpainting, and image super resolution. The tested denoisers are namely Recursive Filter (RF)

*Figure 1.* 12 Testing Images for the Experiment

(Young & Van Vliet, 1995), Total Variation (TV) (Louchet & Moisan, 2011), Non-Local Means (NLM) (Buades et al., 2011), BM3D (Dabov et al., 2007) and Deep Convolutional Neural Network (DnCNN) (Zhang et al., 2018). Note that the DnCNN model that we use is the pretrained model in the MATLAB Deep Learning Toolbox.

We will conduct the experiment as follows. First, we are going to determine which denoiser is the superior one among the ones that we tested. It is important to verify whether we can find one because if not, then there is no superior denoiser, and we cannot verify our hypothesis. Second, we will evaluate the performance of the PnP ADMM solutions using different denoisers, and the implementation is based on Algorithm 1. From that point, we can conclude whether our hypothesis is correct or not, i.e., if a superior denoiser can bring the best performance in the PnP ADMM solution. For the testing images, we will use 12 standard images, which are shown in Figure 1, that are common in image restoration problems to evaluate the performance of PnP ADMM. The codes for the PnP ADMM have been developed in (Chan et al., 2016), and we modify them for our experiment purposes.

### 3.2. First Part of Experiment

As mentioned above, the first part of the experiment is to determine which denoiser is the superior one under our testing environment. The metrics to measure the performance is peak signal to noise ratio (PSNR). The PSNR is defined as an expression for the ratio between the maximum possible power of a signal and the power of distorting noise that affects the quality of its representation. Additionally, the first four denoisers aforementioned are sometimes called deterministic denoisers, which mean that we need to specify the denoising strength $\sigma$. For the ease of presentation, we decide that we will use the parameter $\rho$ to be a constant, i.e., $\rho = 1$, but later we will also do experiment in how changing the value of $\rho$ affects the PnP ADMM solution. Therefore, the remaining parameter needed to compute $\sigma$ is just $\lambda$. For our experiment, we use the following values for $\lambda$:

$$\begin{aligned} \lambda_{RF} &= 0.0005, \\ \lambda_{TV} &= 0.01, \\ \lambda_{NLM} &= 0.005, \\ \lambda_{BM3D} &= 0.001, \end{aligned}$$

For DnCNN, we do not need to specify the value of $\lambda$ since it is already pretrained against all noise levels, assuming the noise level is not too large. We test the performance of the denoisers by using the 12 testing images and adding a zero mean Gaussian noise with standard deviation $\sigma_s$. Here, the noise standard deviation is taken 5 evenly spaced from 0 to 100/255. After that, we calculate the PSNR between the denoised image and the original image. The procedure can be summarized in Algorithm 2.

**Algorithm 2** Denoiser Implementation

**Require:** testing images set $I$, noise level set $\Sigma_s$,$\lambda$
  **for** each image $z$ in $I$ **do**
    **for** each noise level $\sigma_s$ in $\Sigma_s$ **do**
      $y = z + \mathcal{N}(0, \sigma_s^2)$
      **for** each denoiser **do**
        $\hat{x} = D_{\sigma(\lambda)}(y)$
        Compute PSNR between $\hat{x}$ and $z$
      **end for**
    **end for**
  **end for**
  Compute PSNR average over $I$

### 3.3. Second part of experiment

The second part of the experiment mainly deals with performance evaluation of the PnP ADMM solution for image deblurring, inpainting and super resulution, resulted from different denoisers. Same as in the first part, PSNR will be used as an evaluation metric. The procedure is summarized in Algorithm 3. Here, there are some parameters we need to specify, namely $\lambda$, $\rho$, and $tol$. $\lambda$ is already given in the previous section for each denoiser, and $\rho$ and $tol$ are given as

$$\begin{aligned} tol &= 10^{-3}, \\ \rho &= 1. \end{aligned}$$

Note that we use the same $\lambda$ values for each denoiser because we want to do our performance analysis in the same setting, i.e., the parameter $\sigma$, which controls the denoiser strength, is kept the same so that their performance in the PnP ADMM algorithm will reflect their denoiser performance in the first part of the experiment. Additionally, although now the parameter $\rho$ is a constant, we will do an experiment in which $\rho$ is varied (it does not mean that $\rho$ is

**Algorithm 3** PnP ADMM implementation

**Require:** testing images set $I$,noise level set $\Sigma_s$, $\lambda$, $\rho$, tol
  **for** each image $z$ in $I$ **do**
    **for** each noise level $\sigma_s$ in $\Sigma_s$ **do**
      $y = z + \mathcal{N}(0, \sigma_s^2)$
      **for** each denoiser **do**
        Set $k = 1$
        **while** $k \leq maxIter$ and $\Delta_k > tol$ **do**
          $\rho_k = \rho$
          $x^{(k+1)} = \arg\min_x f(x) + \frac{\rho_k}{2}||x - (v^{(k)} - u^{(k)})||^2$
          $v^{(k+1)} = D_{\sigma_k}(x^{(k+1)} + u^{(k)})$
          $u^{(k+1)} = u^{(k)} + (x^{(k+1)} - v^{(k+1)})$
          $\epsilon_1 = ||x^{(k+1)} - x^{(k)}||_2/\sqrt{n}$
          $\epsilon_2 = ||v^{(k+1)} - v^{(k)}||_2/\sqrt{n}$
          $\epsilon_3 = ||u^{(k+1)} - u^{(k)}||_2/\sqrt{n}$
          $\Delta_{k+1} = \epsilon_1 + \epsilon_2 + \epsilon_3$
        **end while**
        Compute PSNR between $x^{(k)}$ and $z$
      **end for**
    **end for**
  **end for**
  Compute PSNR average over $I$

updated at every iteration), and see if the performance of these denoisers are still similar. Readers might notice that the update rule we use in Algorithm 3 is different from that in Algorithm 1. We choose to just use the $\rho$ to be constant because we are not seeking the best performance of the PnP ADMM. Here, we are more interested in comparing the PnP ADMM solutions across different denoisers but in similar settings.

## 4. Experiment Results

### 4.1. Denoiser Performance Experiment

Aforementioned, the first part of the experiment deals with determining the best denoiser among all denoisers we consider in this paper. Figure 2 illustrates the average PSNR of each denoisers at noise levels ranging from 0 to $100/255$ on the 12 testing images. Clearly, we can determine that DnCNN is the best denoiser in this setup since it is able to produce the highest PSNR among all denoisers. The NLM also shows a good performance compared to other deterministic denoisers starting from the noise level $20/255$. On the other hand, it seems that RF and BM3D are the inferior denoisers, especially at higher noise levels.

Additionally, Figure 3 shows the noisy image as well as the denoised images on the noise level $40/255$ on a testing image *Ship512.png*. Visually, the denoised image resulted from the DnCNN has the best quality, where as that resulted from the RF has the lowest quality. The PSNR values for

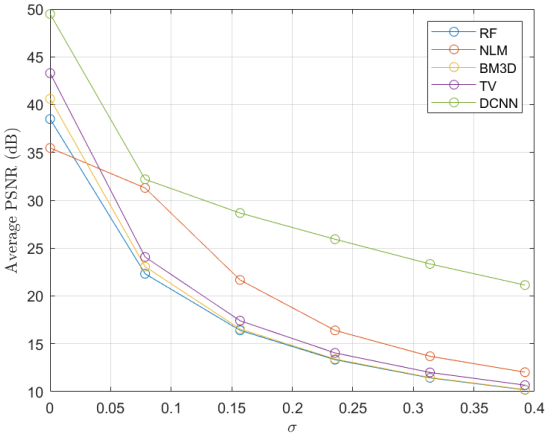


*Figure 2.* PSNR vs noise levels for different denoisers

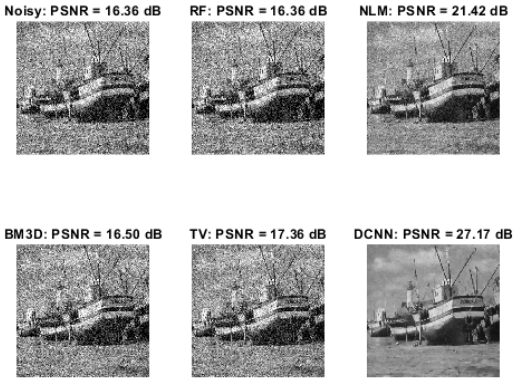


*Figure 3.* Noisy image and denoised images for *Ship512.png*

each noisy and denoised images are also shown. Therefore, we can conclude that the DnCNN is the superior denoiser among all other denoisers we tested.

### 4.2. PnP ADMM Performance Experiment

After evaluating the denoiser performance, we would like to continue our next part of our experiment by plugging the five denoisers we consider in this paper and determining whether the DnCNN produces the best solution in the PnP ADMM algorithm. There are several experiments we will consider. First, let the parameter $\rho$ be 1, i.e. $\rho = 1$.

Before showing the experimental results, we would like to note that for each application, the image goes through some "noising" procedures. For example, for image super-resolution, the image is filtered through a symmetric Gaus-

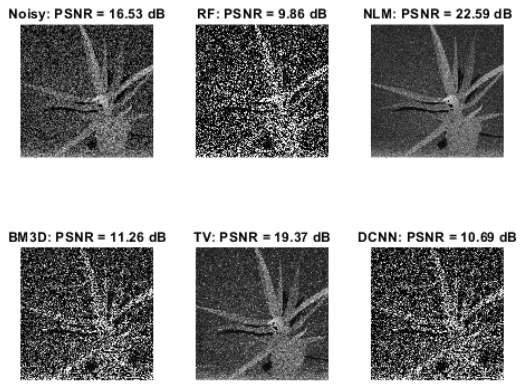


*Figure 4.* Example of PnP ADMM solutions for image deblurring at $\sigma = 40/255$

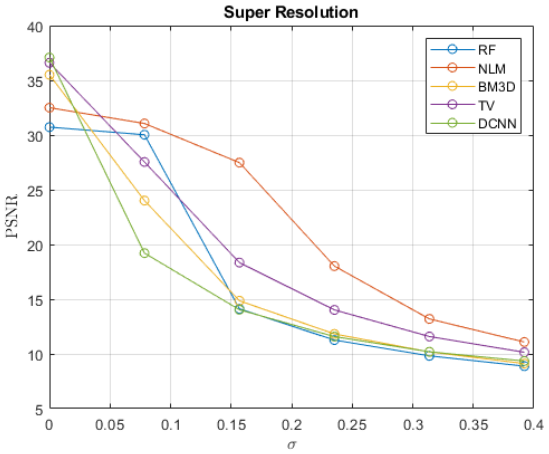


*Figure 5.* PnP ADMM solutions for Image Super-resolution

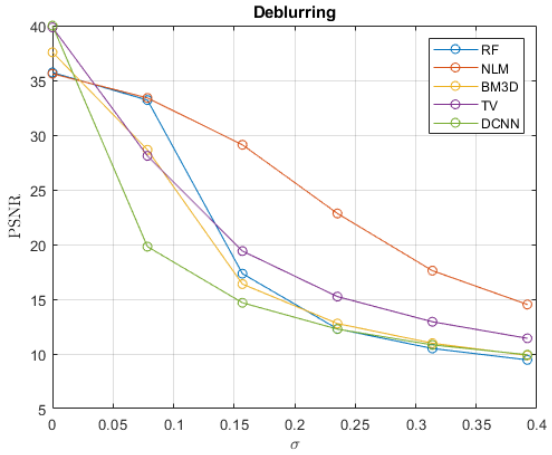


*Figure 6.* PnP ADMM solutions for Image Deblurring

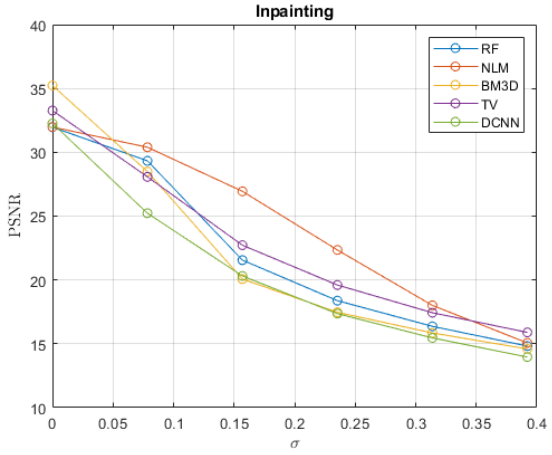


*Figure 7.* PnP ADMM solutions for Image Inpainting

sian lowpass filter of size $9 \times 9$ with standard deviation of 1 and goes through a 2-fold downsampler. For image deblurring, the image goes through the same steps as image super-resolution but with no downsampling. For image inpainting, the image goes through a $20\%$ sampling process. Furthermore, for all applications, the image is added with Gaussian noise at $20/255 - 60/255$ noise levels.

First, we will show you the results of PnP ADMM solutions for deblurring the image *Flower512.png* at the noise level of $40/255$ as it is shown in Figure 4. It is clear that the result from the DnCNN is not the best in terms of deblurring a noisy image. This figure is not the only indication why our initial hypothesis is incorrect. In a more detailed manner, Figure 5, 6, and 7 show the PSNR values of the PnP ADMM solution of the testing image *Flower512.png* at different noise levels with respect to applications in image super-resolution, deblurring, and inpainting, respectively. In all applications, it is clear that the DnCNN does not bring the best PnP ADMM solution, refuting our initial hypothesis. Even, it performs the worst among all other denoisers. Note that the NLM, which is the second best denoiser from our first experiment, actually shows the best performance in all applications of PnP ADMM at higher noise levels. At low noise levels, RF, the worst denoiser, has a higher PSNR value than other deterministic denoisers besides NLM, which is an interesting finding.

Second, we believe that changing the parameter $\rho$ could play an important role on how the PnP ADMM performance changes. As aforementioned, for the sake of simplicity, we only consider the parameter $\rho$ to be a constant. Therefore, the parameters that we can vary in PnP ADMM are

$maxIter$, $tol$ and $\rho$. We believe that there will be no substantial change in varying $maxIter$ and $tol$, and $\rho$ is a more important parameter which acts similar to a regularization parameter. In Figure 8, 9 and 10, we illustrate the PSNR values vs. $\rho$ of the PnP ADMM solutions from each denoiser for some noise levels and for applications in image deblurring, inpainting and super-resolution. Unsurprisingly, the DnCNN again has the worst performance at almost all conditions, except in the image inpainting problem when the noise level is in $20/255$. At that point, the DnCNN actually outperforms other denoisers. Nevertheless, we can see that the NLM gives the best PnP ADMM solution in almost any experiment runs, especially at higher $\rho$ values. Thus, our initial hypothesis is still refuted in this second part of our experiment. Another observation is that, most noticeable in the image deblurring implementation at the noise level $60/255$, the PSNR values of each denoiser are decreasing with the increase of $\rho$. Therefore, it is worth exploring what constitutes the most optimal value of $\rho$ in this case because there is no direct relationship between $\rho$ and the PSNR values.

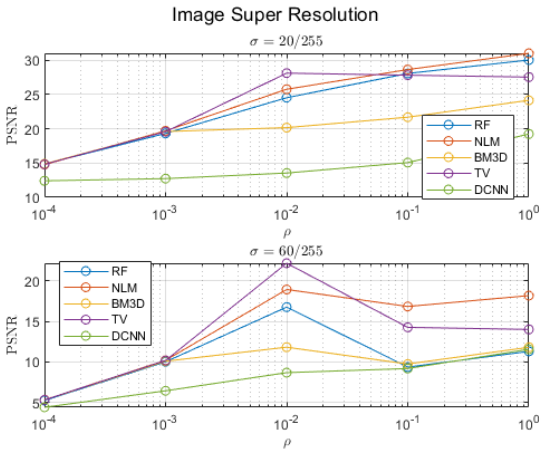


*Figure 8.* PnP ADMM solutions for Image Super-resolution with different values of $\rho$

## 5. Conclusion

In conclusion, this paper has provided some experiments to verify the hypothesis of whether a superior denoiser guarantees the best performance of the PnP ADMM across all noise levels and parameter values. Our first part of the experiments determines that the DnCNN is the best denoiser among all other denoisers we consider at all noise levels. Nevertheless, this particular denoiser does not bring the best PnP ADMM solutions as it is shown on our second part of our experiments. Thus, we can conclude that the best denoiser does not always perform the best in the PnP ADMM

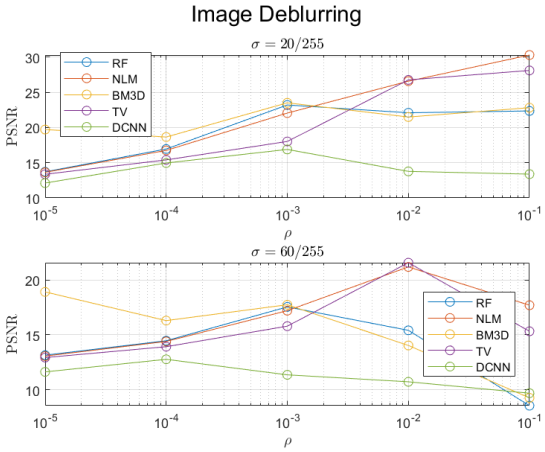


*Figure 9.* PnP ADMM solutions for Image Deblurring with different values of $\rho$

algorithm. Our other finding suggests that different values of $\rho$ generates different results for the PnP ADMM algorithm for each denoiser. Therefore, one way to extend this work is that whether we can find the optimal value of $\rho$ for each denoiser such that it gives the best PnP ADMM solutions. We believe that it is an area that is worth exploring.

## References

Boyd, S., Parikh, N., Chu, E., Peleato, B., and Eckstein, J. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Found. Trends Mach. Learn.*, 3(1):1–122, jan 2011. ISSN 1935-8237. doi: 10.1561/2200000016. URL https://doi.org/10.1561/2200000016.

Buades, A., Coll, B., and Morel, J.-M. Non-local means denoising. *Image Processing On Line*, 1:208–212, 2011.

Chan, S. H., Wang, X., and Elgendy, O. A. Plug-and-play ADMM for image restoration: Fixed point convergence and applications. *CoRR*, abs/1605.01710, 2016. URL http://arxiv.org/abs/1605.01710.

Dabov, K., Foi, A., Katkovnik, V., and Egiazarian, K. Joint image sharpening and denoising by 3d transform-domain collaborative filtering. In *Proc. 2007 Int. TICSP Workshop Spectral Meth. Multirate Signal Process., SMMSP*, volume 2007. Citeseer, 2007.

Dar, Y., Bruckstein, A. M., Elad, M., and Giryes, R. Postprocessing of compressed images via sequential denoising. *IEEE Transactions on Image Processing*, 25(7):3044–3058, 2016. doi: 10.1109/TIP.2016.2558825.

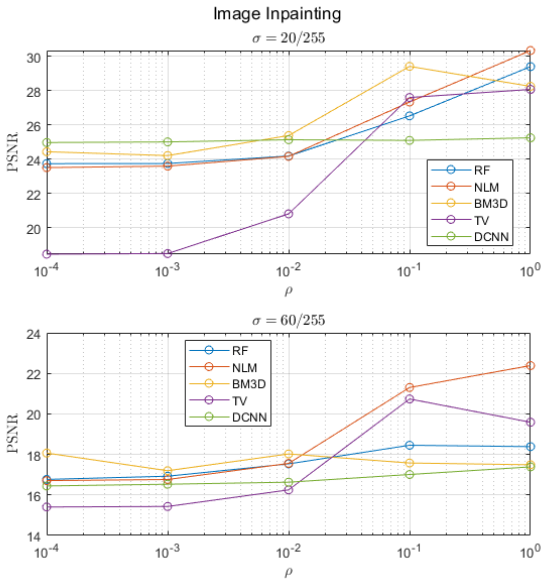


*Figure 10.* PnP ADMM solutions for Image Inpainting with different values of $\rho$

Elharrouss, O., Almaadeed, N., Al-ma'adeed, S., and Akbari, Y. Image inpainting: A review. *Neural Processing Letters*, 51, 04 2020. doi: 10.1007/s11063-019-10163-0.

Guo, Q. Project, 2022. URL https://ece595.qiguo.org/assignments/project.html.

Louchet, C. and Moisan, L. Total variation as a local filter. *SIAM Journal on Imaging Sciences*, 4(2):651–694, 2011.

Romano, Y., Elad, M., and Milanfar, P. The little engine that could: Regularization by denoising (red). *SIAM Journal on Imaging Sciences*, 10(4):1804–1844, 2017. doi: 10.1137/16M1102884. URL https://doi.org/10.1137/16M1102884.

van Ouwerkerk, J. Image super-resolution survey. *Image and Vision Computing*, 24(10):1039–1052, 2006. ISSN 0262-8856. doi: https://doi.org/10.1016/j.imavis.2006.02.026. URL https://www.sciencedirect.com/science/article/pii/S0262885606001089.

Venkatakrishnan, S. V., Bouman, C. A., and Wohlberg, B. Plug-and-play priors for model based reconstruction. In *2013 IEEE Global Conference on Signal and Information Processing*, pp. 945–948, 2013a. doi: 10.1109/GlobalSIP.2013.6737048.

Venkatakrishnan, S. V., Bouman, C. A., and Wohlberg, B. Plug-and-play priors for model based reconstruction. In *2013 IEEE Global Conference on Signal and Information Processing*, pp. 945–948, 2013b. doi: 10.1109/GlobalSIP.2013.6737048.

Young, I. T. and Van Vliet, L. J. Recursive implementation of the gaussian filter. *Signal processing*, 44(2):139–151, 1995.

Zhang, F., Cai, N., Wu, J., Cen, G., Wang, H., and Chen, X. Image denoising method based on a deep convolution neural network. *IET Image Processing*, 12(4):485–493, 2018.

Zhang, K., Ren, W., Luo, W., Lai, W.-S., Stenger, B., Yang, M.-H., and Li, H. Deep image deblurring: A survey, 2022. URL https://arxiv.org/abs/2201.10700.