

## Beyond Gaussian Denoising: Deep CNN Residual Learning for Image Denoising

Shikha Sain

### Abstract

The training of discriminative models for image denoising has recently attracted a lot of interest due to its efficient denoising performance. In this article, we look at the process used to build a direct denoising convolutional neural network (DnCNN). DnCNN has made significant advancements in picture denoising by integrating advancements in extremely deep structures, learning algorithms, and regularisation approaches. Residual learning and packet normalisation are used specifically to speed up learning and improve denoising performance. Blind Gaussian noise may be reduced using our DnCNN approach. This method does not employ additive white Gaussian noise (AWGN), in contrast to existing discriminative noise reduction strategies, which frequently train specific algorithms on AWGN at a certain noise level. In order to implicitly exclude potentially clean images from the buried layer, DnCNN uses a residual learning technique. Due to this property, we were able to perform a variety of common picture denoising tasks, including Gaussian denoising, single image super resolution, and JPEG image denoising. Our thorough testing demonstrates that the DnCNN model can be effectively implemented utilising GPU computation and executes well in a variety of standard picture denoising applications. Gaussian noise reduction and extremely unusual picture resolution. Our thorough testing demonstrates that the DnCNN model can be implemented effectively utilising GPU computation and executes well in a variety of standard picture denoising applications. Gaussian noise reduction and extremely unusual picture resolution. Our thorough testing demonstrates that the DnCNN model can be implemented effectively utilising GPU computation and executes well in a variety of standard picture denoising applications.

**keywords:** Batch normalization, residual learning, convolutional neural networks, image denoising

### 1.Introduction

Image denoising is a long-standing issue that still affects low-level vision in many real-world applications. The objective of image denoising, according to an image degradation model ( $y=x+v$ ), is to recover a clear picture ( $x$ ) from a noisy observation ( $y$ ). One of the often employed estimations is the AWGN, or additive standard deviation of white noise. The pre-modeling of the picture becomes crucial for denoising the image from a Bayesian perspective if the probabilities are known. A variety of models have been employed to replicate past pictures across time, including non-local self-similarity (NSS) models [1, 2, 3, 4], sparse models [4, etc.]. Gradient models [5, 6, 7, 8, 9], Random Field Markov Models (MRF) [10, [11], [12], and gradient models have all been used to do this. For NSS models, contemporary techniques like BM3D [2], LSSC [4], NCSR [6], and WNNM [13] are particularly well-liked. The majority of preimage-based techniques frequently have two important limitations despite their high denoising capabilities. First off, the noise reduction method is arduous since complicated optimisation issues frequently occur during the testing phase of these algorithms [6], [13]. Therefore, it is difficult for classical algorithms to attain great performance without losing computational efficiency. Second, the noise reduction performance is enhanced by the models' frequent non-convexity and huge number of manually chosen parameters. In recent years, we have advanced beyond conventional methods by training previous models using streamlined inference techniques. The testing phase iterative optimisation procedure may be eliminated by the created model. Schmidt and Roth [14] devised the contractile fields (CSF) cascade method. suggested and coupled a model using extended half-quadratic optimisation and random fields. Integrate a single learning framework with algorithms. Chen et al. have suggested a trainable nonlinear response diffusion (TNRD) model. [15, 16] By using a specific number of gradient descent inference steps,

discover the prerequisites for the improved discipline image [12]. The following studies are related: [17], [18]. Although CSF and TNRD make positive strides towards bridging the gap between computational effectiveness and blur quality, their effectiveness is intrinsically constrained by preset types. In particular, the priors employed in CSF and TNRD are restricted in that they fully encompass the structural features of a picture while being based on analytical models. Along with certain manually specified parameters, the parameters are also acquired by gradual greedy learning and cooperative modification at all stages. Blind picture denoising has drawbacks, one of which being the development of a certain pattern at a specific noise level. In this study, picture denoising—or, more specifically, the following method of eliminating noise from noisy images—is taken into account as a direct differential learning problem. We employ a closed-loop convolutional neural network (CNN) as opposed to previously created models that can discriminate between explicit pictures. The usage of CNN is advantageous for three reasons. Very deep architecture CNNs significantly increase the effectiveness of utilising image characteristics [19]. Among the notable developments in regularisation and learning techniques are the Linear Rectifier Unit (ReLU), batch normalisation, and residual learning. to train for CNN.

This speeds up learning and improves 2CNN denoising performance. Third, because CNNs are well suited for parallel processing, they may be used to speed up execution speed on today's powerful GPUs. DnCNN is the abbreviation for the convolutional neural network for noise reduction. Instead of calculating the denoised image  $x$  directly, the proposed DnCNN makes an effort to predict the afterimage  $v$ , which is the difference between the actual values of the noisy observations and the latent appropriate image. In other words, the proposed DnCNN's hidden layer procedure mistakenly deletes potentially good pictures. The effectiveness of DnCNN training is then stabilised and enhanced using the batch normalisation technique. Batch normalisation and residual learning are discovered to work effectively together. Despite the fact that the aim of this work is to improve the efficiency of a Gaussian denoiser, we have discovered that the image degradation model of a Gaussian denoiser may be changed to a single super high image resolution (SISR). (The difficulty in investigating bicubic oversampling of low-resolution images and the disparity between the original image and the original high-resolution image. The same picture degradation model that considers both the original image and the JPEG may be used to explain the JPEG image deblocking issue. SISR and JPEG picture deblocking can be viewed in this context as two particular illustrations of "common" image noise reduction issues. Even compared to AWGN, noise and deblocking in SISR and JPEG are substantially different. It seems sense to wonder if a CNN model can be trained to handle such a diverse variety of photo denoising issues. Gaussian denoising, SISR, and JPEG image denoising are only a few of the popular image denoising jobs that DnCNN is capable of handling by analysing the link between TNRD and DnCNN [16]. I suggest extending research that has demonstrated that DnCNN learns at particular noise levels. Modern techniques like WNNM [13], BM3D [2], and TNRD [16] are less effective than gaussian noise reduction. For Gaussian blind blur at an unknown noise level, DnCNN with a single model outperforms BM3D [2] and TNRD [16] trained on a particular noise level. Applying DnCNN to a variety of well-known picture denoising tasks appears to produce promising results. Additionally, for three typical picture denoising tasks—JPEG denoising with varying quality factors, SISR with various scaling factors, and blind Gaussian noise reduction—we utilised a single DnCNN model. A list of the contributions to this work is provided below: 1) We suggest an end-to-end trainable deep CNN for gaussian denoising. In contrast to earlier systems based on a deep neural network that evaluated possibly clean images directly, this network employs residual learning techniques to distinguish potentially clean photos from noisy observations. 2) We discovered that residual learning and batch normalisation may both be significantly enhanced. By accelerating the denoising and training processes, this increases the effectiveness of CNN training. When there is some noise, DnCNN surpasses cutting-edge Gaussian denoising methods in terms of quantitative measures and visual quality. 3) DnCNN handles typical picture denoising jobs with ease. We train a DnCNN model to extract particular noise for blind Gaussian denoising. It could outperform similar technologies that are appropriate for your

level. The three primary picture denoising tasks of blind Gaussian denoising, SISR, and JPEG denoising should be manageable by a single DnCNN model. The remaining material of this article is divided into the next sections. In Section II, a synopsis of relevant work is presented. In Section III, the suggested DnCNN model is initially introduced. After that, it is improved for overall image noise reduction. Extensive tests are provided to assess DnCNN in Section IV. Finally, Section IV offers thorough tests for DnCNN evaluation. Finally, Section IV offers thorough tests for DnCNN evaluation.

## 2. Literature review

### A. Image denoising with deep neural networks

Deep neural networks have made various attempts to solve the problem of noise reduction. Convolutional neural networks (CNN) have been proposed in [23] by Jain and Seung for use in image denoising. They said that CNNs have the same, if not more, expressive power than MRF models. [24] succeeded in removing noise from images using a multi-layer perceptron (MLP). In [25], Gaussian noise suppression was addressed using a summation sparse denoising auto-encoder approach, yielding results similar to K-SVD [5].

Two methods based on deep neural networks, MLP and TNRD, can compete with BM3D and show promising performance. However, special models are trained for certain noise levels in MLP [24] and TNRD [16]. To our knowledge, the development of CNN for global image denoising has not yet been explored.

### B. Party normalization and rest training

Convolutional neural network systems have recently shown excellent performance in processing a variety of visual tasks, because to the availability of access to large datasets and developments in deep learning techniques. Revised linear units of measurement (ReLU) [20], depth-width tradeoffs [19], [26], variable initiation [27], gradient-based optimisation techniques [28], [29], [30], normalization batches [21]), and residual learning [22] are all outstanding instances of advancements in CNN model learning. The effectiveness of CNNs is influenced by a number of other variables, including the performance of training on modern, powerful GPUs.

**1) Residual learning:** For CNN, residual learning [22] was initially offered as a solution to the performance degradation issue, i.e., the fact that even learning accuracy degrades as network depth rises. increase. Because the residual network is considerably simpler to learn than the initial unreferenced mapping, it purposefully creates several stacked layers of residual mappings. Using a residual learning strategy, it is quite simple to train a very deep CNN, which increased the accuracy of object recognition and picture categorization [22].

Residual learning is also included in the suggested DnCNN model. Unlike residual networks [22], which employ numerous residual units (i.e. ID tags), our DnCNN only employs a single residual unit for residual picture prediction. The goal of the residual learning formulation becomes more apparent when the link between it and TNRD [16] is examined, and it may be expanded to address many typical picture denoising issues. Numerous low-level visual issues, like single-image superresolution [31] and colour picture demosaicing [32], have employed residual prediction techniques prior to the creation of residual networks [22].

**2) Batch normalization:** The CNN model is trained using stochastic gradient descent (SGD) in mini-batches.

The SGD mini-batch is a simple and efficient algorithm, but internal covariant shifting [21], i.e. changing the range of the internal nonlinear input during training, greatly reduces the efficiency of training. Batch normalization [21] includes a normalization step, a scaling step, and an offset step before correcting nonlinearity at each level. This reduces internal covariate shifts. Batch normalization only introduces two parameters and we can update them with backpropagation. The benefits of batch normalization include faster training, better performance, and less sensitive initialization. See [21] for more information on batch normalization.

Batch normalization research for CNN-based image denoising has generally not been done yet. We have found empirically that the combination of batch normalization and residual training produces more efficient training and better denoising results.

### 3. Proposed CNN denoising model

In this part, we suggest an extension of the CNN blurring model to solve a variety of common photo denoising tasks. Creating the network architecture and training the algorithm using training data are typically the two steps in properly training a CNN model for a particular job. We adapt the VGG network design [19] for image denoising from the perspective of network architecture design, and we determine the network's length using the ideal patch size applied to modern denoising techniques. We combine batch normalisation with the remaining train equation to train the model quickly while enhancing denoising performance.

#### A. Network depth

By changing the size of the convolutional filter to 3 3, we implement the concept of [19] and eliminate all pool layers.

As a result, at the depth  $d=d$ , the receptive field of the DnCNN must be  $(2d+1)(2d+1)$ . The size of the receptive field must be enlarged in order to utilise context information in significant portions of the image.

You may discover the ideal balance between performance and efficiency by properly choosing the DnCNN depth. It is a crucial component in architectural planning.

According to studies [23, 24], the noise reduction method's effective spot size corresponds to the noise reduction neural network's response field size. Additionally, bigger effective patch sizes are typically needed for greater noise levels in order to gather more contextual data for recovery [34]. We examine the effective patch sizes of several widely used denoising techniques by setting noise level = 25 in order to guide the technical design of DnCNN. As a result of our adaptive search for widows of size 25 25 twice, the final effective patch size for BM3D [2] is 4949. The real patch size is substantially greater (361,361) since WNNM [13] uses huge search windows to conduct non-local iterative searches similar to BM3D. Because we first use a patch of size 3939 to create the predicted patch and then apply a filter of size 99 to average the output patches, the actual patch size for MLP [24] is 4747. A five-phase CSF and TNRD are composed of 10 convolutional layers with a  $7 \times 7$  filter size and  $61 \times 61$  effective patch size, respectively [14] and [16]. Finally, to average the output patches, we use a filter with a size of 99. A five-phase CSF and TNRD are composed of 10 convolutional layers with a  $7 \times 7$  filter size and  $61 \times 61$  effective patch size, respectively [14] and [16]. Finally, to average the output patches, we use a filter with a size of 99. A five-phase CSF and TNRD are composed of 10 convolutional layers with a  $7 \times 7$  filter size and  $61 \times 61$  effective patch size, respectively [14] and [16].

Table 1 displays the effective patch sizes employed by the various techniques at noise level = 25. As can be seen, EPLL [33] has an effective patch size of 3636, which is the smallest. It will be fascinating to see if first-order noise reduction techniques can compete with DnCNN, which has a receptive field similar to EPLL. To set the Gaussian noise reduction to a certain noise level, set the DnCNN receptive field size to 35 35 and the associated depth to 17. Use a broader receptive field and a depth of 20 for performing other typical picture noise reduction activities.

#### B. Networking

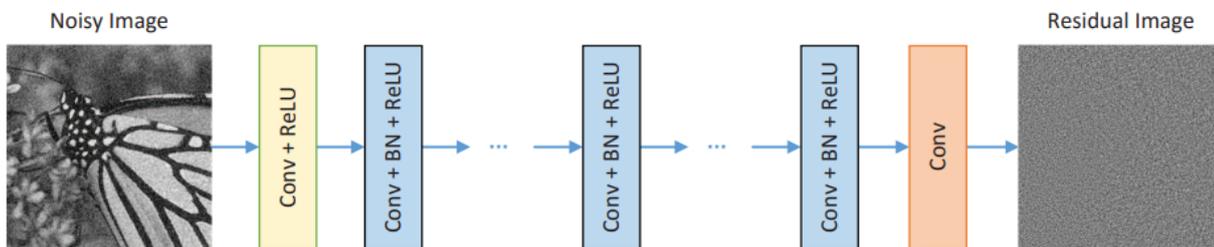
DnCNN will take noisy observations with the input of  $y=x+v$ . In order to predict potentially clean images, discriminative denoising models like MLP [24] and CSF [14] attempt to train the mapping function  $F(y) = x$ . To train the residual mapping  $R(y) \vee$  DnCNN, we use the residual learning formula. As a result,  $x = yR(y)$ . Alternately, the variation in root mean square between the anticipated afterimage and the resultant image from the noisy input.

$$\ell(\Theta) = \frac{1}{2N} \sum_{i=1}^N \|\mathcal{R}(y_i; \Theta) - (y_i - x_i)\|_F^2 \quad (1)$$

It can be used as a loss function to form DnCNN formable parameters. where '(y<sub>i</sub>, x<sub>i</sub>)' denotes N pairs of pure training images (patches) with noise, where N<sub>i</sub>=1. The proposed DnCNN architecture for R(y) training is shown in Figure 1. The DnCNN architecture and methods for minimizing edge artifacts are described below.

Table 1- The data table below shows the effective patch sizes used by various image denoising methods when the noise level, denoted sigma (σ), is 25.

method	Effective patch size
BM3D [2]	49×49
VNNM [13]	361×361
EPLL [33]	36×36
MLP [24]	47×47
CRL [14]	61×61
TNRD [16]	61×61



**Figure 1- DnCNN Network Architecture**

**1. Deep DnCNN Architecture:** There are three different kinds of layers in DnCNN.

(i) Conv+ReLU: This is the network's bottom layer. Put 64 3x3xc filters to use. where c stands for the quantity of picture channels (1 for grayscale images and 3 for colour images). In order to account for nonlinearities, the output of the convolution operation is sent through a rectified linear unit (ReLU) process.

Where  $D$  is the network depth, the hidden layers 2 to  $(D-1)$  are represented by (ii) Conv+BN+ReLU. These levels make use of  $64 \times 3 \times 3 \times 64$  filters. To stabilise, speed up, and improve denoising performance, batch normalisation (BN) is applied after the process of convolution (Conv) and ReLU procedures.

(iii) Conv: This network layer uses  $3 \times 3 \times 64$  C filters to reassemble the output. The lowest layer is it.

The DnCNN model has two crucial components. One employs batch normalisation to speed learning and improve performance, while the other uses a residual learning equation to learn  $R(y)$ , or the leftovers of the noisy picture  $y$ . denoising in performance. DnCNN separates the structure of an image from noisy input using hidden layers. Other algorithms, such as EPLL and WNNM, also use iterative denoising techniques that are analogous to this one. On the other hand, DnCNN is taught from scratch.

2. Reducing edge artefacts: Edge artefacts may develop because many low-level vision applications require that the output image equal the input image in size. DnCNN uses a zero padding strategy before to each layer's convolution operation to ensure that each feature map is exactly the same size as the input image. This simple zero padding method doesn't produce edge artefacts because to the powerful DnCNN features. Zero padding may cause certain problems, but these can be reduced by denoising and the model's excellent ability to learn intricate patterns.

### C. Image denoising using residual learning and batch normalization

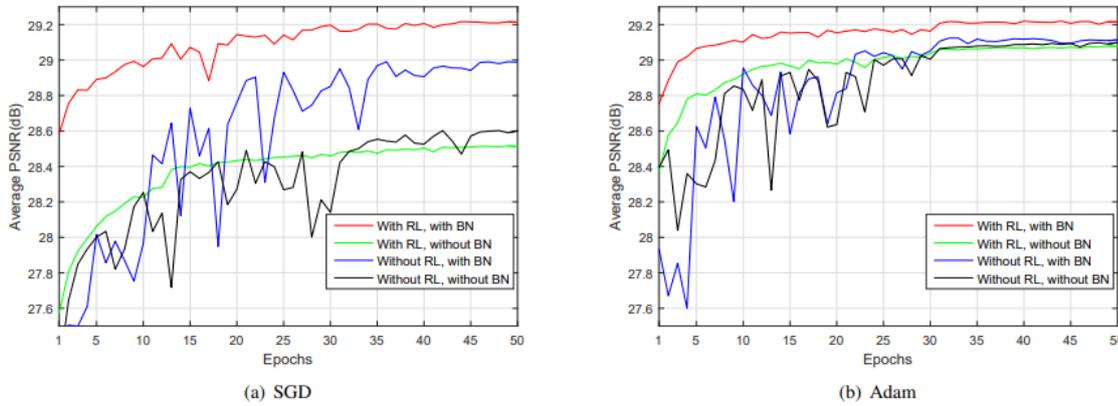
The network in Figure 1 may be used to train either the original mapping  $F(y)$  for the prediction  $x$  or the residual mapping  $R(y)$  for the prediction  $v$ . According to [22], providing beginning values substantially simplifies the optimisation of the residual mapping. Identity mapping is quite similar to mapping. Keep in mind that, especially at low noise levels, the noisy observation  $y$  resembles a latent brilliant picture  $x$  more so than an afterimage  $v$ .

Figure 2 displays the average PSNR values for these two training algorithms using the identical gradient-based optimisation technique and network architectural parameters with and without batch normalisation. The Adam algorithm and the stochastic momentum gradient descent (SGD) algorithm are two optimisation techniques that rely on the gradient [30]. First, we discover that learning with residual learning rather than learning with initial mapping might result in a faster and more dependable convergence. On the other hand, without batch normalisation, basic residual creation with conventional SGD cannot match with contemporary denoising approaches like TNRD (28.92 dB).

We blame subpar performance on internal covariate changes [21] brought on by training-induced adjustments to network parameters. In order to rectify, batch normalisation is utilised. Then, we can observe that the batch-normalized original mapping training (blue line) converges more slowly and performs worse at denoising than the residual mapping training (red line). The SGD and Adam optimisation algorithms provide the highest performance for networks with residual learning and batch normalisation in particular. In other words, Gaussian distributions are actually connected to batch normalisation and Gaussian afterimage noise reduction. Remaining training and batch normalisation are anticipated to work well together for Gaussian denoising 1. This conclusion may be strengthened by the analysis that follows. The residual learning without batch normalisation (green line) converges more quickly than the residual learning with batch normalisation (red line), but it still performs badly, as illustrated in Figure 2.

Batch normalisation, however, results from residual learning. Batch normalisation has a detrimental effect on convergence in the absence of residual training, as illustrated in Figure 2 (blue line). To expedite training and enhance performance, batch normalisation can be employed in conjunction with residual training (red line). There are only a certain number of pictures in each minivan (for instance, 128). Without residual training, the distribution of input data for a layer is influenced by the pictures included in each training mini-batch as well as the relationship between the input data's intensity and the convolutional features and their surrounding regions. DnCNN operates at the integration layer and implicitly filters out potentially clean pictures using residual learning. As a result, each layer's input has a Gaussian-like distribution and is less correlated with the information contained in the images.

In conclusion, by speeding and stabilising the learning process, the use of batch normalisation in conjunction with residual training can enhance denoising performance.



**Figure 2 displays the results of two gradient-based optimisation techniques—(a) SGD and (b) Adam—for the reduction of Gaussian noise for four distinct models. At a noise level of 25, four unique models were trained using different residual learning (RL) and batch normalisation (BN) settings. For the purpose of reviewing our findings, we use 68 natural pictures from Berkeley's segmentation dataset.**

**D. Connection to TNRD**

Another way to think of our DnCNN is as a one-step TNRD technique [15], [16]. The TNRD's general objective is to teach discerning answers to the following inquiries:

The objective function that must be minimised in the TNRD mathematical model has the following form:

$$\Psi(y - x) + \lambda \sum_{k=1}^k \sum_{p=1}^n \rho_k((f_k * x)_p) \quad (2)$$

Here,

y - noisy image

x - denoised image

N - image size

λ is the regularization parameter

f<sub>k</sub> - k-th filter

ρ<sub>k</sub> is the kth penalty function

For Gaussian noise reduction,  $\Psi(z) = 1/2 \|z\|^2$ .

The first step in the iterative denoising process based on this model is the gradient descent step given by

$$x_1 = y - \alpha \lambda \sum_{k=1}^k (f_k * \varphi_k(f_k * y)) - \alpha \partial \Psi(z) / \partial z |_{z=0} \quad (3)$$

In the above formula

α - step size

φ<sub>k</sub> is the derivative of the penalty function (also called the impact function).

f<sub>k</sub> — attached filter f<sub>k</sub>

For Gaussian noise reduction,  $\partial \Psi(z) / \partial z |_{z=0} = 0$ . Substituting this into equation (3), we get:

$$v_1 = y - x_1 = \alpha \lambda \sum_{k=1}^k (f_k * \varphi_k(f_k * y)) \quad (4)$$

where v<sub>1</sub> is the estimated remainder of x with respect to y.

DnCNN is a generalization of the one-step TNRD model that modifies the model by replacing the rectified linear unit (ReLU) influence function to increase network depth and allow batch normalization.

Understanding the use of residual learning in CNN-based picture reconstruction requires connecting to TNRD. Since most of the parameters in equation (4) are generated from the previous analytic term in equation (2), the majority of the DnCNN parameters represent prior pictures.

Equation (3) can still be used to obtain  $v_1$  for non-Gaussian noise distributions or unknown levels of Gaussian noise for  $\partial\Psi(z)/\partial z |_{r=0} = 0$ . This is true for many types of noise distribution. For example, the generalized Gaussian distribution.

DnCNN performs reliably even in afterimage prediction, even with complex noise distribution, gradually removing potentially clean images in the hidden layer.

### **E. General improvement of image noise reduction**

For a given noise level, every discriminative Gaussian noise reduction technique now in use, including MLP, CSF, and TNRD, produces a unique model [16], [24]. A usual method is to estimate the noise floor first, train a model with the correct noise floor, and then use that model to reduce the unknown noise using a Gaussian noise model. As a result, the effectiveness of noise reduction is influenced by the noise estimation's accuracy. Additionally, several techniques with non-Gaussian noise distribution cannot be applied, including SISR and JPEG deblocking.

The DnCNN's potential for extensive picture denoising is revealed by the examination of Section III-D. DnCNN is first upgraded to get rid of Gaussian noise with an undetermined noise level. Create noisy photos with a variety of noise levels (such [0.55]) to train a single DnCNN model. Test pictures with noise levels that are within acceptable ranges may be denoised using a trained DnCNN model without the noise level being inferred.

By creating customised models for a number of widely used photo denoising applications, we enhance DnCNN. JPEG deblocking, SISR, and blind Gaussian noise reduction are the three main tasks that we concentrate on. We employ AWGN pictures with different noise levels, downsampled images with different scale values, and JPEG images with different quality factors to train a DnCNN model.

According to experimental data, a single trained DnCNN model may offer remarkable performance for any of the three typical picture denoising tasks.

## **4. Experimental results**

### **A. Experimental context**

1. Training and test data: In accordance with [16], we employ 400, 180, and 180 training pictures for Gaussian noise reduction with known or unknowable noise levels. We discovered that adding more training data marginally enhanced performance. We take into consideration three noise levels, namely = 15, 25 and 50, to train a DnCNN for Gaussian noise reduction with known noise levels. We cut 128 1600 patches of size 40 40 to train the model. DnCNN-S is the name of the model DnCNN for Gaussian noise reduction with a specified noise level.

For blind Gaussian denoising training, we used a DnCNN model with a patch size of 50/50 and a noise level range of [0.55]. To train the model, 128,3000 patches are cut. DnCNN-B is the sole DnCNN model for reducing Gaussian blind noise.

To fully evaluate the test photographs, we use two different sets of test data. One is made up of the 12 images from Figure 3 and the other is from the Berkeley Segmentation Dataset (BSD68), which consists of 68 photographs of the natural world [12]. Noting that none of these images are from the training dataset, it should be noted that they are all often used to evaluate Gaussian noise reduction strategies.

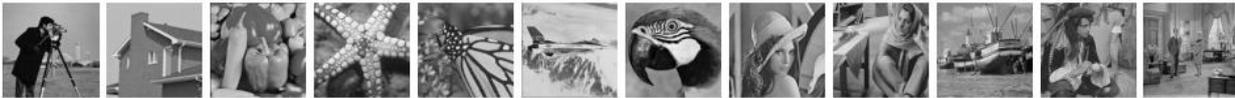
We also develop a denoising model for color-blind (CdnCNNB) pictures, in addition to grayscale ones. For training, we utilise the remaining 432 colour photos from the Berkeley segmentation dataset, and for testing, we use the colour versions of the BSD68 dataset. With a noise threshold of [0, 55], 128,3000 patches are trimmed to sizes between 50 and 50.

To create a single model for three popular picture denoising tasks, we generated 200 training images from the Berkeley segmentation dataset and 91 similar images from [36]. Make use of the data set we've supplied. Add a consistent amount of Gaussian noise between [0.55] to a picture to make it look noisy. At first, we bicubic-scale the high-resolution image by 1, and then by 2, 3, and 4. The SISR input is produced in this way. To generate the JPEG

deblocking input, we compress the picture with a quality factor between 5 and 99 using the MATLAB JPEG encoder. All of these images are used to train a DnCNN model. A total of 1,288,000 picture pairings of 50 by 50 pixels are made for the structure. In mini-batch training, patch pairs are manipulated via rotations and flips. The parameters are set up with DnCNN-B. DnCNN-3 is a single model that can handle all three of these frequent picture denoising jobs. In Section IV-E, we explain in depth how we evaluate DnCNN-3 by generating separate test sets for each task. In Section IV-E, we explain in depth how we evaluate DnCNN-3 by generating separate test sets for each task. In Section IV-E, we explain in depth how we evaluate DnCNN-3 by generating separate test sets for each task. Learning how to set up a network We trained the DnCNN-S network to a depth of 17, the DnCNN-B network to a depth of 20, and the DnCNN-S network to a depth of 20 to get an adequate amount of spatial information for noise reduction. Find the residual mapping  $R(y)$  using the loss function in equation (1) to make a prediction about the residual  $v$ . Use 1). Using the method given in [27], we use SGD to generate the initial weights; the weight reduction is 0.0001, the momentum is 0.9, and the minilot size is 128. In order to train our DnCNN model, we run it through 50 iterations. The learning rate went from  $1e1$  to  $1e4$  after 50 epochs, a drop of an exponential order.

### II. Comparison of approaches

The suggested DnCNN methodology is divided into four components: one generation method (EPLL [33]), two non-local comparison methods (WNNM [2] and WNNM [13]), three divergent approaches to learning (MLP [24], CSF [14], and TNRD [16]), and two non-local comparison techniques (BM3D [2] and WNNM [13]). Be aware that the GPU implementations CSF and TNRD offer excellent efficiency while still producing good image quality.



In fig. 3 shows 12 commonly used test photos.

Table 2. Best results are in bold. The DnCNN-S model performs best at all noise levels ( $\sigma=15, 25, 50$ ), followed by DnCNN-B at  $\sigma=50$ . is not available. Tested at specified noise levels.

Noise level ( $\sigma$ )	BM3D	VNNM	EPLL	MLP	FSC	TNRD	DnCNN-S	DnCNN-B
15	31.07	31.37	31.21	-	31.24	31.42	31.73	31.61
25	28.57	28.83	28.68	28.96	28.74	28.92	29.23	29.16
50	25.62	25.87	25.67	26.03	-	25.97	26.23	26.23

### III. Quantitative and qualitative assessment

The average PSNR findings for the BSD68 dataset as determined by various techniques are displayed in Table II. As can be observed, DnCNN-S and DnCNN-B can perform better in terms of PSNR than rival technologies. Comparing the MLP and TNRD approaches to the standard BM3D method, the PSNR is improved by roughly 0.35 dB. Few techniques exceed BM3D by more than 0.3 dB on average, according to [34], [38]. In every one of the noise levels,

the DnCNN-S models beats the BM3D by 0.6 dB. Surprisingly, even with a single model having an unknown noise level, our DnCNN-B outperforms rival algorithms trained for the same known noise level. = 50, we find that DnCNN-S and DnCNN-B perform roughly 0.6 dB better than BM3D. The PSNR limit (0.7 dB) for BM3D anticipated in [38] is extremely close to being reached in this case.

**Table 3.- Shows the PSNR (peak signal-to-noise ratio) in dB (decibels) for various noise reduction methods on 12 different test images.**

picture	BM3D	VNNM	EPLL	MLP	FSC	TNRD	DnCNN-S	DnCNN-B
he man	31.91	32.17	31.85	-	31.95	32.19	32.61	32.10
Accommodation	34.93	35.13	34.17	-	34.39	34.53	34.97	34.93
pepper	32.69	32.99	32.64	-	32.85	33.04	33.30	33.15
Starfish	31.14	31.82	31.13	-	31.55	31.75	32.20	32.02
monal.	31.85	32.71	32.10	-	32.33	32.56	33.09	32.94
Airlines companies	July 31	31.39	31.19	-	31.33	31.46	31.70	31.56
parrot	31.37	31.62	31.42	-	31.37	31.63	31.83	31.63
Lena	34.26	34.27	33.92	-	34.06	34.24	34.62	34.56
Barbaric	33.10	33.60	31.38	-	31.92	32.13	32.64	32.09
boat	32.13	32.27	31.93	-	32.01	32.14	32.42	32.35
Male	31.92	32.11	32.00	-	32.08	32.23	32.46	32.41

pair	32.10	32.17	31.93	-	31.98	32.11	32.47	32.41
mean	32.37	32.70	32.14	-	32.32	32.50	32.86	32.68

Table 3 shows the PSNR results of the different methods for the 12 test images in Figure 3. The bold for each noise level in each image shows the best PSNR result. It is clear that the proposed DnCNN-S produces the most images with the highest PSNR. In particular, DnCNN-S outperforms competing approaches by 0.2 to 0.6 dB for most photographs, but for two images consisting mostly of repeating structures: "House" and "Barbara". This conclusion is in agreement with the conclusions of [39]. Approaches based on discriminant learning often work best for images with irregular textures, while methods based on non-local averaging generally work best for images with regular and repetitive textures. In fact,

Figures 4 and 5 show the visual results of some of the methods.

**TABLE IV. COMPARISON OF WORKING TIMES OF DIFFERENT NOISE REDUCTION METHODS**

Method	Image size	BM3D	VNNM	EPLL	MLP	CSF (CPU/GPU)	TNRD (CPU/GPU)	DnCNN-S (CPU/GPU)	DnCNN-B (CPU/GPU)
	256×256	0.65	203.1	25.4	1.42	2.11/-	0.45/0.010	0.74 / 0.014	0.90 / 0.016
	512×512	2.85	773.2	45.5	5.51	5.67 / 0.92	1.33 / 0.032	3.41/0.051	4.11/0.060
	1024×1024	11.89	2536.4	422.1	19.4	40.8/1.72	4.61/0.116	12.1/0.200	14.1/0.235

Naturally, BM3D, WNNM, EPLL, and MLP often yield edges and textures that are overly smooth. Although TNRD is prone to artefacts in smooth regions, it preserves fine detail and sharp edges. DnCNN-S and DnCNN-B, on the other hand, are able to not only recover tiny details and sharp edges but also provide beautiful effects in smooth regions.

Figures 1 and 2 provide a visual comparison of CDnCNN-B with the CBM3D benchmark for colour picture noise reduction. It is evident that CDnCNN-B can recover a picture with more accurate colour information, but CBM3D occasionally results in misleading colour artefacts. Additionally, CDnCNN-B can create pictures with crisper edges and more details when compared to CBM3D.

Figure 7 illustrates the average improvement in PSNR for BM3D/CBM3D for various noise levels using the DnCNN-B/CDnCNN-B models. As demonstrated, for various noise levels, our DnCNN-B/CDnCNN-B models consistently beat BM3D/CBM3D in a considerable way. A single DnCNN-B network may be trained to perform blind Gaussian noise reduction at various noise levels, according to the experiment's findings.

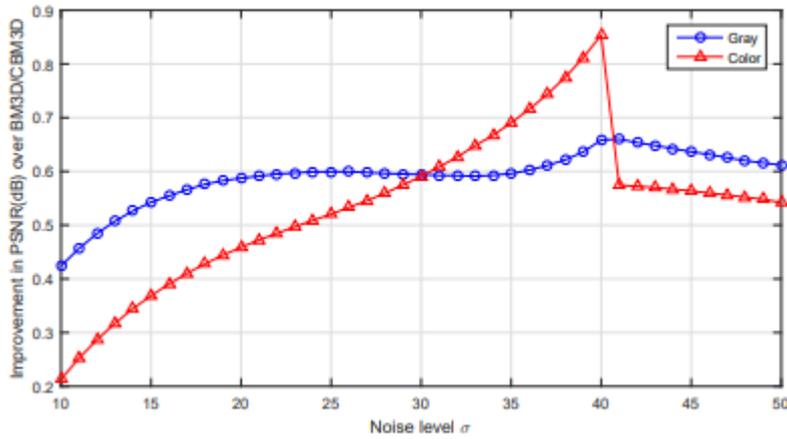


Figure 8 shows the average increase in PSNR obtained by DnCNN-B/CDnCNN-B compared to BM3D/CBM3D at different noise levels. Results are evaluated using BSD68 grayscale and color datasets.

Table V: Average PSNR/SSIM results for different methods of single-frame super-resolution, Gaussian noise reduction, and JPEG image deblocking.

Gaussian Removal	Noise		BM3D	TNRD	DnCNN-3
database	noise level		PSNR/SSIM	PSNR/SSIM	PSNR/SSIM
BSD68	15		31.08 / 0.8722	31.42 / 0.8826	31.46 / 0.8826
	25		28.57 / 0.8017	28.92 / 0.8157	29.02 / 0.8190
	50		25.62 / 0.6869	25.97 / 0.7029	26.10 / 0.7076

Super single image resolution			TNRD	WDSR	DnCNN-3
Database	multiplier		PSNR/SSIM	PSNR/SSIM	PSNR/SSIM

set 5	2	36.86 / 0.9556	37.56 / 0.9591	37.58 / 0.9590
	3	33.18 / 0.9152	33.67 / 0.9220	33.75 / 0.9222
	4	30.85 / 0.8732	31.35 / 0.8845	31.40 / 0.8845
set 14	2	32.51/0.9069	33.02 / 0.9128	33.03 / 0.9128
	3	29.43 / 0.8232	29.77 / 0.8318	29.81 / 0.8321
	4	27.66 / 0.7563	27.99 / 0.7659	28.04 / 0.7672
BSD100	2	31.40 / 0.8878	31.89 / 0.8961	31.90 / 0.8961
	3	28.50 / 0.7881	28.82 / 0.7980	28.85 / 0.7981
	4	27.00 / 0.7140	27.28 / 0.7256	27.29 / 0.7253
Urban 100	2	29.70 / 0.8994	30.76 / 0.9143	30.74 / 0.9139

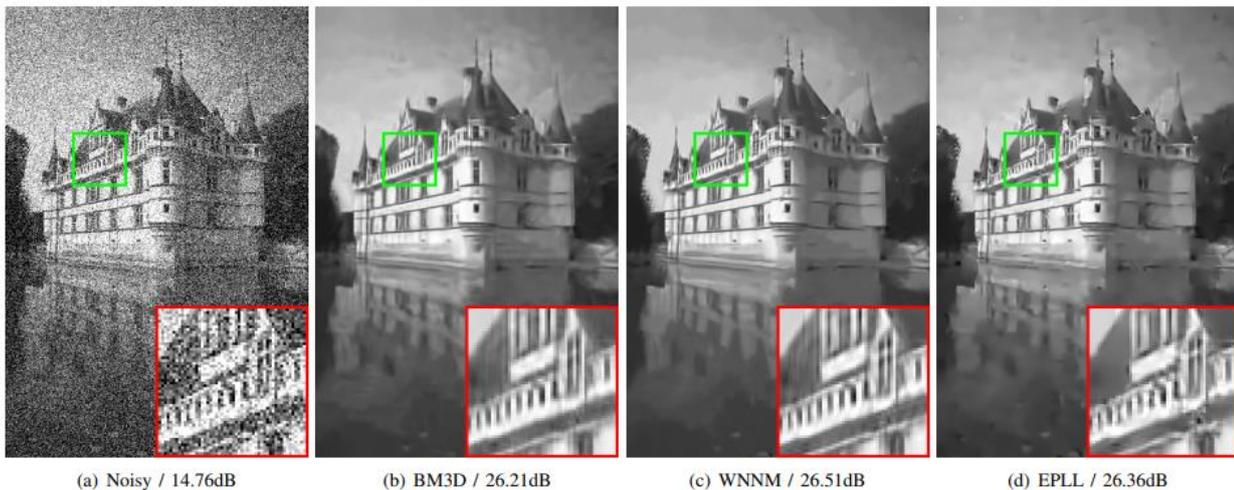
	3	26.42 / 0.8076	27.13 / 0.8283	27.15 / 0.8276
	4	24.61/0.7291	25.17 / 0.7528	25.20 / 0.7521

Unlock JPEG images		AP-CNN	TNRD	DnCNN-3
Database	quality factor	PSNR/SSIM	PSNR/SSIM	PSNR/SSIM
classic 5	10	29.03 / 0.7929	29.28 / 0.7992	29.40 / 0.8026
	20	31.15 / 0.8517	31.47 / 0.8576	31.63 / 0.8610
	30	32.51/0.8806	32.78 / 0.8837	32.91 / 0.8861
	40	33.34 / 0.8953	-	33.77/0.9003
LIVE1	10	28.96 / 0.8076	29.15 / 0.8111	29.19 / 0.8123
	20	31.29 / 0.8733	31.46 / 0.8769	31.59/0.8802
	30	32.67 / 0.9043	32.84 / 0.9059	32.98 / 0.9090
	40	33.63 / 0.9198	-	33.96 / 0.9247

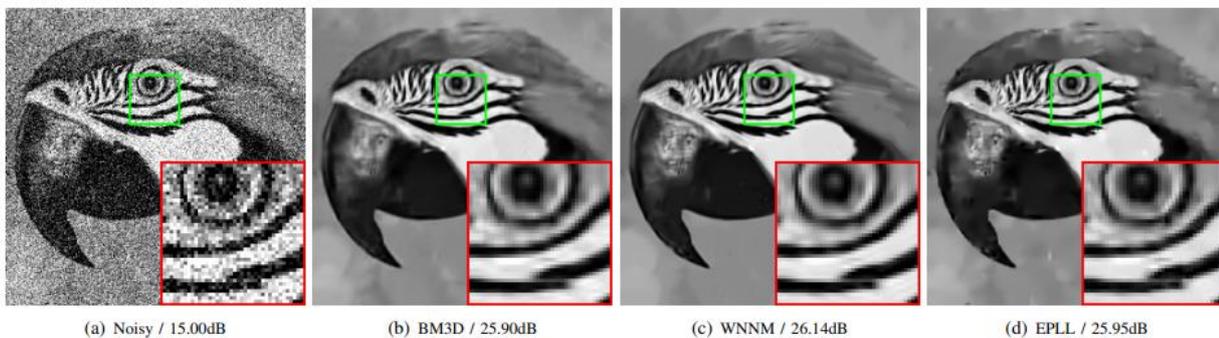
**IV. Distance**

The speed at which an image restoration approach may be evaluated is as important to its visual quality. The processing durations for several denoise techniques are shown in Table IV for images with sizes of 256, 256, 512,

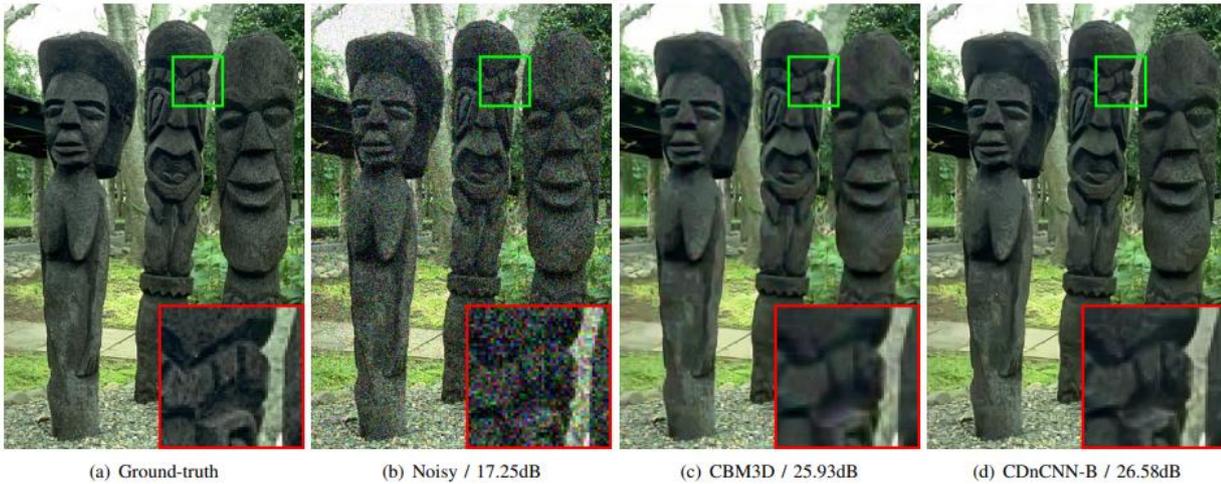
512, and 1024x1024 at a noise level of 25. The CSF, TNRD, and DnCNN algorithms are perfectly suited for parallel processing on this platform, hence we also provide corresponding GPU runtimes. Using the Nvidia cuDNNv5 machine learning package, we speed the calculation of the supplied DnCNN on the GPU. As in [16], the time used for data transmission between the CPU and GPU is not taken into consideration. It is clear that the proposed DnCNN can run exceptionally quickly on computers and is faster than both of the distinguishing models, MLP and CSF. Even while our DnCNN is quicker than BM3D and TNRD, it is still highly competitive on CPU solutions because to the improved picture quality. The proposed DnCNN delivers incredibly alluring computational efficiency for GPU time. For instance, cleaning up noise from 512,512 photos with an unknown noise level requires 60 ms. This definitely benefits TNRD.



In fig. Figure 4 shows the result of denoising a single BSD68 image at a noise level of 50.



In fig. Figure 5 shows the image of the parrot after denoising with a noise level of 50.



In fig. Figure 6 shows the results of denoising a single frame color image from DSD68 data at a noise level of 35.

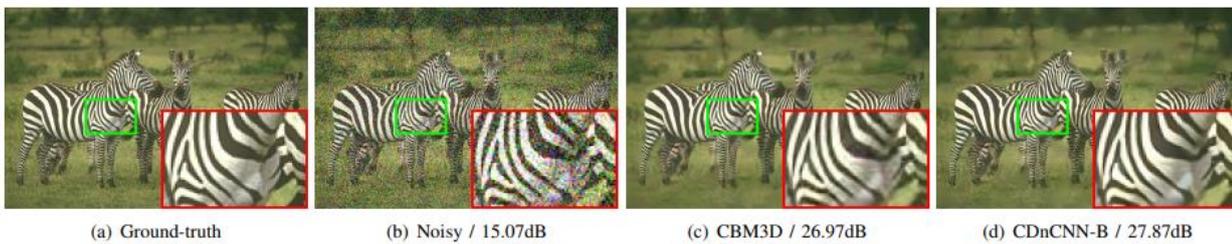


Figure 7. Result of the color image denoising operation on the DSD68 dataset image with a noise level of 45

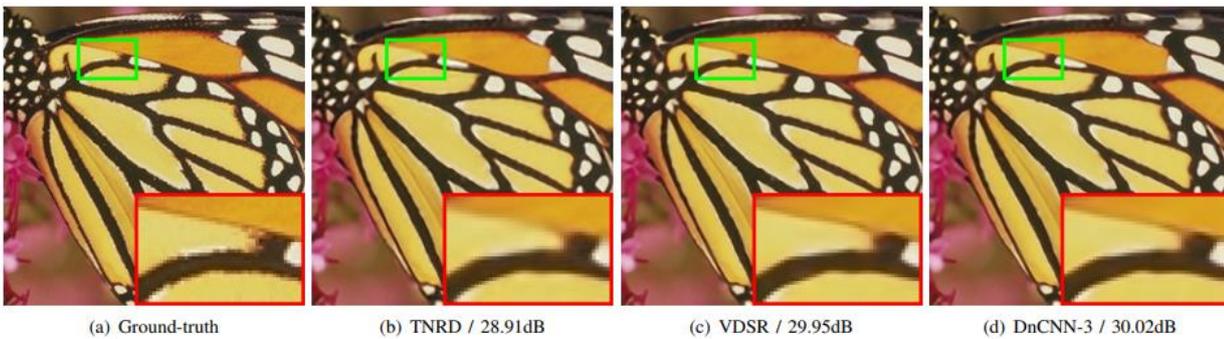


Figure 9. Ultra-high resolution result of a single butterfly image in the Set5 dataset using 3x scaling.

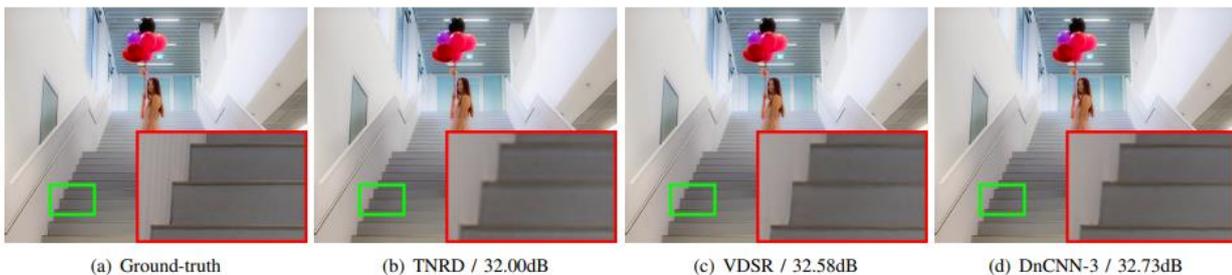


Figure 10. Ultra-high resolution results for a single image using the Urban100 dataset and a scale factor of 4

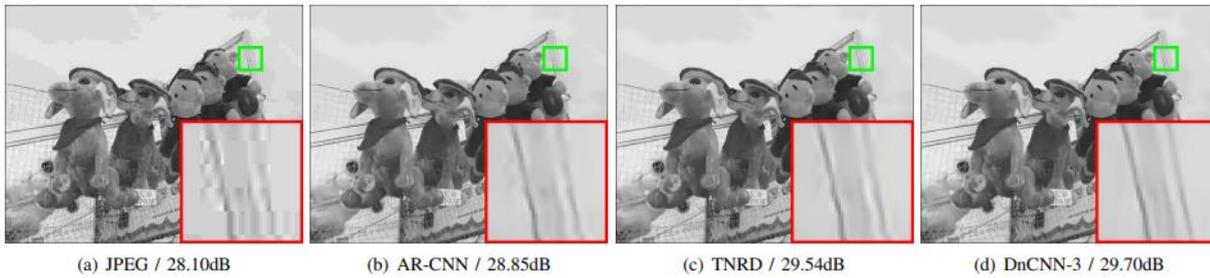


Figure 11 shows the results of deblocking a "Carnivaldolls" JPEG image using the LIVE1 dataset and a quality factor of 10.

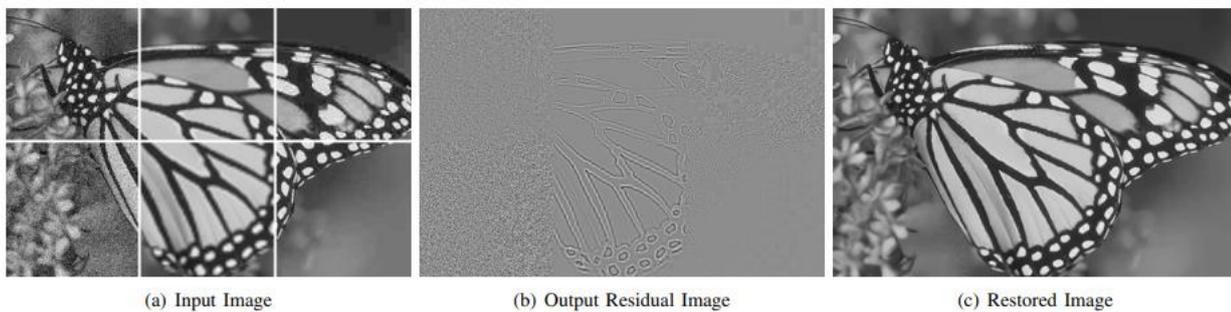


Figure 12 illustrates how our method addresses three distinct jobs. The input pictures are made up of low-resolution bicubic interpolated images, noisy images, and JPEG images with quality factors of 10 (top right) and 30 (bottom right), respectively. Images at zoom factors 2 and 3 in the centre, as well as a poor quality image at zoom factor 2, are shown. Note that there is no afterimage; the white lines in the inputted image are just utilised to divide the picture into its six portions. for viewing, normalised to the [0, 1] range. The image has been repaired, and it seems natural and devoid of flaws.

### V. An Attempt to Train a Single Model to Address Three Typical Image Noise Reduction Issues

To highlight the potential of the suggested DnCNN model, a DnCNN-3 model was trained on three typical picture denoising tasks, including blind Gaussian denoising, SISR, and JPEG image denoising. As far as we're aware, there isn't currently a solution that unifies these three responsibilities into a single model. Therefore, for each job, we contrast DnCNN-3 with pertinent state-of-the-art methods. Below are the comparative plan and assessment information for each position.

- Modern BM3D and TNRD algorithms are evaluated for reducing Gaussian noise. The BSD68 dataset is used to evaluate performance. You are probably already aware of how noisy BM3D and TNRD are.
- We take into account TNRD and VDSR as two revolutionary innovations for the SISR [35]. While TNRD trained distinct models for each scale component, VDSR [35] taught a single algorithm for all three scale elements (i.e., 2, 3, and 4). The four test cases used in [35] were BSD100, Urban100, Set5 and Set14.
- Two cutting-edge JPEG picture deblocking techniques, TNRD [16] and AR-CNN [41], are contrasted with our DnCNN-3. We trained four distinct models with JPEG quality parameters of 10, 20, 30, and 40, respectively, using the AR-CNN technique. On JPEG quality factors of 10, 20, and 30 in TNRD, three models were trained. We employ the Classic5 and LIVE1 test datasets, much like in [41].

Table V displays the mean PSNR and SSIM results for several notable noise reduction projects.

As can be shown, even when training a single model for three distinct tasks, the DnCNN-3 model beats the blinded TNRD and BM3D in terms of Stochastic noise reduction. For the SISR, it is significantly higher than the TNRD and comparable to the VDSR. DnCNN-3 performs better than TNRD and AR-CNN when it comes to of PSNR while deblocking JPEG pictures by within 0.1 dB and 0.3 dB, respectively.

Figures 9 and 10 show a visual comparison of several HSIS tactics. It is clear that DnCNN-3 and VDSR have the ability to provide sharp edges and fine details, in contrast to TNRD, which usually has curved lines and soft edges. The results of numerous JPEG deblocking methods are shown in Figure 11. As you can see, DnCNN-3 can fix crooked lines whereas TNRD and AR-CNN are more likely to produce straight lines in the first place. A alternate representation of the possible results of the recommended paradigm is shown in Figure 12. It is evident that DnCNN3 may provide beautiful and enticing results even when the input image has varying levels of distortion.

## Conclusion

In order to discriminate between noise and noisy observation, this research introduced a robust convolutional neural network (CNN) for denoising images. Batch normalising and residue training are coupled to hasten training and enhance denoising performance. In contrast to conventional discriminant techniques that train several models for various noise levels, our single DnCNN model can handle the blind Random blur with known noise levels. Three common denoising of images tasks—stochastic blur with uncertain levels of noise, a single picture high-resolution with variable scaling factors, and JPEG picture deblocking with various quality parameters—can all be handled by a single DnCNN model. According to results of comprehensive testing, the suggested technique delivers positive picture blur effectiveness in terms of both quantity and quality and also has a potential performance via GPU execution.

## Reference

- [1] A. Bouades, B. Col, J.-M. Morel, "Nonlocal Image Denoising Algorithms", IEEE Conference on Computer Vision and Pattern Recognition, Volume 2, 2005, 60–65 pages.
- [2] K. Dabov, A. Foy, V. Katkovnik, and K. Yeghiazaryan, "Image Noise Reduction Using Cooperative Filtering of Sparse 3D Transform Domains", IEEE Transactions on Image Processing, Volume 16, no. 8, pp. 2080–2095, 2007.
- [3] A. Bouades, B. Col, J.-M. Morel, "Non-Local Imaging and Film Denoising", International Journal of Computer Vision, volume 76, no. 2, p. 123–139, 2008
- [4] J. Mairal, F. Bach, J. Ponce, G. Sapiro and A. Zisserman, "Nonlocal sparse models for image reconstruction", IEEE International Conference on Computer Vision, 2009, pp. 2272-2279.
- [5] M. Elad and M. Aaron, "Image Noise Reduction with Sparse and Redundant Representations in Trained Vocabularies", IEEE Transactions on Image Processing, volume 15, no. 12, pp. 3736–3745, 2006.
- [6] W. Dong, L. Zhang, G. Shi, and S. Li, "Unlocated Sparse Representations for Image Reconstruction", IEEE Transactions on Image Processing, volume 22, no. 4, p. 1620–1630, 2013.
- [7] LI Rudin, S. Osher and E. Fatemi, "Nonlinear Total Variation Denoising Algorithm", Physica D: Nonlinear Phenomena, vol. 60, no. 1, p. 259–268, 1992.
- [8] S. Osher, M. Burger, D. Goldfarb, J. Xu, and W. Ying, "Iterative Regularization Method for Total Variation-Based Image Restoration", Multiscale Modeling and Simulation, Volume 4 , no. 2, p. 460–489, 2005.
- [9] Y. Weiss and WT Freeman, "What Makes a Good Natural Image Model?" IEEE Conference on Computer Vision and Pattern Recognition, 2007, pp. 1-8.

- [10] X. Lan, S. Roth, D. Hattenlocher and MJ Black, "Efficient Belief Propagation with Learned Higher-Order Markov Random Fields", European Conference on Computer Vision 2006, pp. 269-282.
- [11] SZ Lee, Markov simulation of random fields in image analysis. Springer Science & Business Media, 2009.
- [12] S. Roth and MJ Black, The Realm of Professionals, International Journal of Computer Vision, volume 82, no. 2, p. 205–229, 2009
- [13] Gu X., Zhang L., Zuo W., and Feng X., "Kernel-Weighted Norm Minimization with Application to Image Noise Reduction", IEEE Conference on Computer Vision and Pattern Recognition forms, 2014, pp. 2862–2869.
- [14] W. Schmidt and S. Roth, "Compression fields for efficient image reconstruction", IEEE Conference on Computer Vision and Pattern Recognition, 2014, pp. 2774-2781.
- [15] Yu. Chen, V. Yu and T. Pok, "On the study of optimized reaction-diffusion processes for efficient image reconstruction", IEEE Conference on Computer Vision and Pattern Recognition, 2015 , 5261–5269 pages.
- [16] Y. Chen and T. Pok, "Teachable Nonlinear Reactive Diffusion: A Flexible Framework for Fast and Efficient Image Reconstruction", to be published in 2016 IEEE Transactions on Pattern Analysis and Machine Intelligence.
- [17] W. Schmidt, K. Rother, S. Novozin, J. Janksari and S. Roth, "Discriminative non-blind deblurring", IEEE Conference on Computer Vision and Pattern Recognition, 2013, p. 604-611.
- [18] W. Schmidt, J. Janksari, S. Novozin, S. Roth and C. Rother, "Regression Tree Field Cascades for Image Reconstruction", IEEE Conference on Computer Vision and Pattern Recognition, volume 38, no. 4, p. 677–689, 2016.
- [19] K. Simonyan and A. Zisserman, "Extremely Deep Convolutional Networks for Large-Scale Image Recognition", International Conference on Learning and Representations, 2015.
- [20] A. Krizhevsky, I. Sutskever and GE Hinton, "Imagenet Classification with Deep Convolutional Neural Networks", Advances in Neural Information Processing Systems, 2012, p. 1097-1105.
- [21] S. Ioffe and K. Szegedy, "Batch Normalization: Accelerating Network Deep Learning by Reducing Internal Covariate Changes," International Conference on Machine Learning 2015, p. 448-456.
- [22] K. He, S. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 770-778.
- [23] V. Jain and S. Seung, Natural Image Denoising with Convolutional Networks, Advances in Neural Information Processing Systems, 2009, p. 769-776.
- [24] HC Burger, CJ Schuler, S. Harmeling, "Noise reduction in images: can simple neural networks compete with BM3D? » 2012 IEEE Conference on Computer Vision and Pattern Recognition, p. 2392-2399.
- [25] J. X [25] J. Xie, L. Xu and E. Chen, "Noise reduction and image coloring with deep neural networks", Advances in Neural Information Processing Systems, 2012, pp. 341–349.

- [26] V. Lehtinen, J. Munkberg, J. Hasselgren, S. Lane, T. Karras, M. Aittala, T. Ayla, "Noise2Noise: Learning to retrieve an image without own data", International Conference on Learning automatic, 2018. , pages 2965-2974.
- [27] K. Zhang, W. Zuo, and L. Zhang, "FFDNet: Towards a Fast and Flexible CNN-Based Image Denoising Solution," IEEE Transactions on Image Processing, vol. 27, no. 9, p. 4608–4622, 2018.
- [28] C. Tomasi and R. Manducci, "Two-way filtering of gray and color images", 6th International Conference on Computer Vision, 1998, p. 839-846.
- [29] P. Milanfar, "A Journey Through Modern Image Filtering: New Practical and Theoretical Ideas and Methods", IEEE Signal Processing, volume 30, no. 1, p. 106–128, 2013.
- [30] Y. Chen, X. He, J. Yang, Q. Wu, "Improving Image Noise Reduction with Deep Texture Prea", IEEE Transactions on Image Processing, volume 30, no. 1, p. 200-210, 2020.
- [31] S. Guo, Z. Yang, Q. Zhang, W. Zuo, L. Zhang, "Towards convolutional blind noise reduction of real photographs", IEEE Conference on Computer Vision and Pattern Recognition, 2019 , 1712–1722 pages.
- [32] X. Xu, Q. Yan, Y. Xia, J. Jia, "Deriving structure from textures by relative complete reshaping", ACM Transactions on Graphics, volume 31, no. 6, 2012.
- [33] G. Ghiazi, H. Lee, M. Kudlur, V. Dumoulin, J. Schless, "Real-time study of the structure of artistic stylization arbitrary neural networks", Proceedings of the UK Machine Vision Conference, 2017 .
- [34] AG Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, H. Adam, "MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications", preprint arXiv, 2017.
- [35] C. Szegedy, W. Vanhooke, S. Joffe, J. Schless, Z. Warina, "Rethinking the Initial Architecture of Computer Vision", IEEE Conference on Computer Vision and Pattern Recognition, 2016 , p. 2818–2826.
- [36] M. Tan, KV Le, "EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks", International Conference on Machine Learning, 2019, pp. 6105-6114.
- [37] Yu. Zhang, K. Li, K. Li, L. Wang, B. Zhong, and Yu. Fu, "Deep Learning Image Denoising Algorithms: A Review", Neurocomputing, Volume 323, 37 -50 Pages, 2019.
- [38] X. Gu, L. Zhang, W. Zuo and X. Feng, "Deep convolutional network learning for ultra-high resolution images", Pattern Recognition, 2015, pp. 184-193.
- [39] J. Kim, J. Kwon Lee, and K. Mu Lee, "Precise Ultra-High Image Resolution Using Very Deep Convolutional Networks", IEEE Conference on Computer Vision and Pattern Recognition, 2016, p. 1646-1654.
- [40] X. Mao, C. Shen and Y. Yang, "Image reconstruction using a convolutional autoencoder with symmetric traversals", Advances in Neural Information Processing Systems, 2016, pp. 2802-2810.
- [41] S. Wang, D. Liu and Z. Zhang, "Deep Networks for Super-Resolution Imaging Using Sparse Priors", IEEE International Conference on Computer Vision, 2015, pp. 370-378.



- [42] L. Zhang, L. Zhang and H. Shen, “Ultra-efficient noise reduction models based on convolutional neural networks with composite activation functions”, *Journal of Imaging and Graphics*, vol.1, no. 1, p. 20–28, 2023.
- [43] M. Garbi, G. Chaurasia, S. Paris, F. Duran, “Dehicking and deep collaborative denoising”, *ACM Transactions on Graphics*, vol.35, no. 6, 2016.