LLNet: A Deep Autoencoder Approach to Natural Low-light Image Enhancement

Kin Gwn Lore, Adedotun Akintayo, Soumik Sarkar

Iowa State University, Ames IA-50011, USA

Abstract

In surveillance, monitoring and tactical reconnaissance, gathering visual information from a dynamic environment and accurately processing such data are essential to making informed decisions and ensuring the success of a mission. Camera sensors are often cost-limited to capture clear images or videos taken in a poorly-lit environment. Many applications aim to enhance brightness, contrast and reduce noise content from the images in an on-board real-time manner. We propose a deep autoencoder-based approach to identify signal features from low-light images and adaptively brighten images without over-amplifying/saturating the lighter parts in images with a high dynamic range. We show that a variant of the stacked-sparse denoising autoencoder can learn from synthetically darkened and noise-added training examples to adaptively enhance images taken from natural low-light environment and/or are hardware-degraded. Results show significant credibility of the approach both visually and by quantitative comparison with various techniques.

Keywords:
image enhancement, natural low-light images, deep autoencoders

1. Introduction and motivation

Good quality images and videos are key to critical automated and human-level decision-making for tasks ranging from security applications, military missions, path planning to medical diagnostics and commercial recommender systems. Clean, high-definition pictures captured by sophisticated camera systems

Email addresses: kglore@iastate.edu (Kin Gwn Lore), akintayo@iastate.edu (Adedotun Akintayo), soumiks@iastate.edu (Soumik Sarkar)

provide better evidence for a well-informed course of action. However, cost constraints often limit large scale applications of such systems. Thus, relatively inexpensive sensors are used in many cases. Furthermore, adverse conditions such as insufficient lighting (e.g. low-light environments, night time) worsen the situation. As a result, many areas of application, such as Intelligence, Surveillance and Reconnaissance (ISR) missions (e.g. recognizing and distinguishing enemy warships), unmanned vehicles (e.g. automated landing zones for UAVs), and commercial industries (e.g. property security, personal mobile devices) stand to benefit from improvements in image enhancement algorithms.

Recently, deep learning (DL)-based approaches gained immense traction as they are shown to outperform other state-of-the-art machine learning tools in many computer vision applications, including object recognition [1], scene understanding [2], occlusion detection [3], prognostics [4], and policy reward learning [5]. While neural networks have been widely studied for image denoising tasks, we are not aware of any existing works using deep networks to both enhance and denoise images taken in poorly-lit environments. In the present work, we approach the problem of contrast enhancement from a representation learning perspective using deep autoencoders (what we refer to as Low-light Net, or LLNet) that are trained to learn underlying signal features in low-light images and adaptively brighten and denoise. The method takes advantage of the local patch-wise contrast improvement similar to the works in [6] to enhance contrast such that the improvements are done relative to local neighbors to prevent over-amplifying the intensities of already brightened pixels. Furthermore, the same neural network is trained to learn the structures of objects that persist through noise in order to produce a brighter, denoised image.

Contributions: The present paper presents a novel application of using a class of deep neural networks–stacked sparse denoising autoencoder (SSDA)–to enhance natural low-light images. To the best of the author's knowledge, this is the first application of using a deep architecture for (natural) low-light image enhancement. We propose a training data generation method by synthetically modifying images available on Internet databases to simulate low-light environments. Two types of deep architecture are explored - (i) for simultaneous learning of contrast enhancement and denoising (LLNet) and (ii) sequential learning of contrast-enhancement and denoising using two modules (*staged* LLNet or S-LLNet). The performances of the trained networks are evaluated and compared against other methods on test data with synthetic noise and artificial darkening. Performance evaluation is repeated on natural low-light images to demonstrate the enhancement capability of the synthetically trained model applied on a realistic set of images obtained with regular cell-phone camera in low-light environments. Hidden layer weights of the deep network are visualized to offer insights to the features learned by the model. Another contribution is that the framework performs *blind* contrast enhancement without requiring a reference image frame (e.g. using information from a previous frame in video enhancement [7], and the use of daytime counterparts [8]), which is absolutely vital in scenarios where new environments are frequently encountered (e.g. in tactical reconnaissance).

2. Related work

There are well-known contrast enhancement methods such as improving image contrast by histogram equalization (HE) and its variants such as contrast-limiting adaptive HE (CLAHE) brightness preserving bi-HE (BBHE) and quantized bi-HE (QBHE) [9, 10, 11, 12]. Subsequently, an optimization technique, OCTM [13] was introduced for mapping the contrast-tone of an image with the use of mathematical transfer function. However, this requires weighting of some domain knowledge as well as an associated complexity increase. Available schemes also explored using non-linear functions like the gamma function [14] to enhance image contrast.

Image denoising tasks have been explored using BM3D [15], K-SVD [16], and non-linear filters [17]. Using deep learning, authors in [18] presented the concept of denoising autoencoders for learning features from noisy images while [19] applied convolutional neural networks to denoise natural images. In addition, authors in [20] implemented an adaptive multi-column architecture to robustly denoise images by training the model with various types of noise and testing on images with arbitrary noise levels and types. Stacked denoising autoencoders were used in [21] to reconstruct clean images from noisy images by exploiting the encoding layer of the multilayer perceptron (MLP).

Fotiadou et al. [22] enhanced natural low-illumination images using sparse representations of low-light image patches in an appropriate dictionary to approximate the corresponding daytime images. Dong et al. [7] proposed an algorithm that inverts the dark input frames and performs de-hazing to improve the quality of the low light images. A related method is presented in [23] involving dehazing algorithms. Another technique, proposed in [8], separates the image into two components–reflectance and illuminance–and enhance the images using the reflectance component. The separation of the components, however, is difficult; therefore it may introduce unwanted artifacts in the reconstructed images.

Perhaps one of the most challenging tasks is to gather sufficiently large dataset of low-light images to train the deep learning model. The NORB object recognition dataset [24] contains natural images taken at 6 different illumination levels, but the limited size of the training set is insufficient for training. With this motivation, we also propose a method of simulating low-light environments by modifying images obtained from existing databases.

3. The Low-light Net (LLNet)

The proposed framework is introduced in this section along with training methodology and network parameters.

3.1. Learning features from low-light images with LLNet

SSDAs are sparsity-inducing variant of deep autoencoders that ensures learning the invariant features embedded in the proper dimensional space of the dataset in an unsupervised manner. Early proponents [18] have shown that by stacking several denoising autoencoders (DA) in a greedy layer-wise manner for pretraining, the network is able to find a better parameter space during error back-propagation.

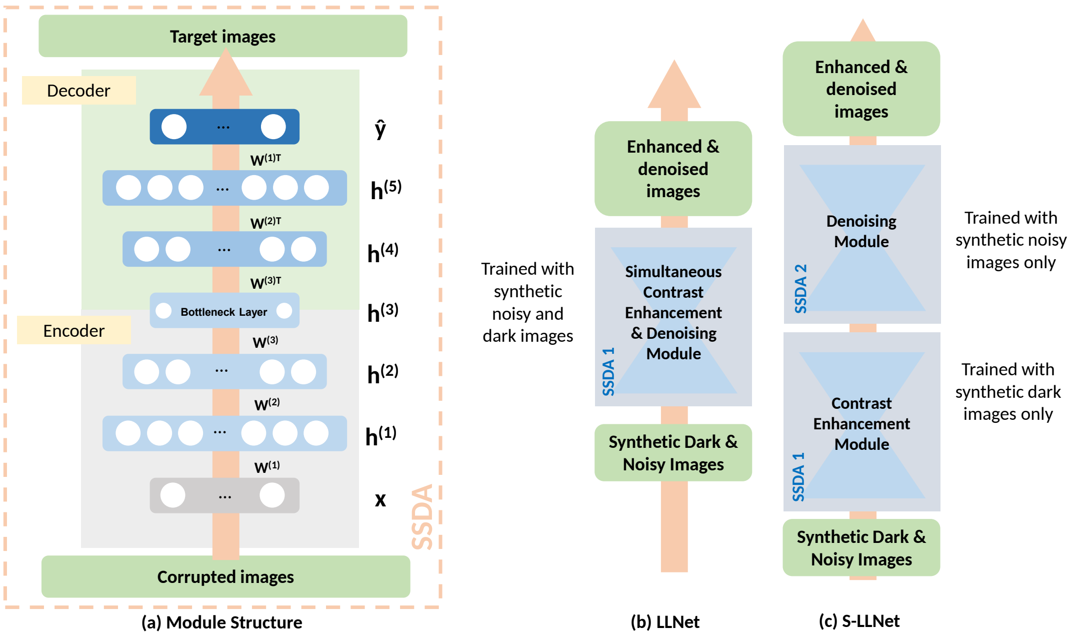
Let $\mathbf{y} \in \mathcal{R}^N$ be the clean, uncorrupted data and $\mathbf{x} \in \mathcal{R}^N$ be the corrupted, noisy version of $\mathbf{y}$ such that $\mathbf{x} = \mathbf{My}$, where $\mathbf{M} \in \mathcal{R}^{N \times N}$ is the high-dimensional, non-analytic matrix assumed to have corrupted the clean data. With DA, feed-forward learning functions are defined to characterize each element of $\mathbf{M}$ as follows:

$$h(\mathbf{x}) = \sigma(\mathbf{Wx} + \mathbf{b})$$

$$\hat{\mathbf{y}}(\mathbf{x}) = \sigma'(\mathbf{W}'\mathbf{h} + \mathbf{b}')$$

where σ and σ' denote the encoding and decoding functions (either of which is usually the sigmoid function $\sigma(\mathbf{s})$ or $\sigma'(\mathbf{s}) = (1 + \exp(-\mathbf{s}))^{-1}$ of a single DA layer with K units, respectively. $\mathbf{W} \in \mathcal{R}^{K \times N}$ and $\mathbf{b} \in \mathcal{R}^K$ are the weights and biases of each layers of encoder whereas $\mathbf{W}' \in \mathcal{R}^{N \times K}$ and $\mathbf{b} \in \mathcal{R}^K$ are the weights and biases for each layer of the decoder. $h(\mathbf{x}) \in \mathcal{R}^K$ is the activation of the hidden layer and $\hat{\mathbf{y}}(\mathbf{x}) \in \mathcal{R}^N$ is the reconstruction of the input (*i.e.* the output of the DA).

LLNet framework takes its inspiration from SSDA whose sparsity-inducing characteristic aids learning features to denoise signals. In the present work, we take the advantage of SSDA's denoising capability and the deep network's complex modeling capacity to learn features underlying in low-light images and produce enhanced images with minimal noise and improved contrast. A key aspect to


Target images
Decoder
ŷ
W(1)T
h(5)
W(2)T
h(4)
W(3)T
Bottleneck Layer
h(3)
Encoder
W(3)
h(2)
W(2)
h(1)
W(1)
x
SSDA
Corrupted images
Trained with synthetic noisy and dark images
Enhanced & denoised images
Simultaneous Contrast Enhancement & Denoising Module
SSDA 1
Synthetic Dark & Noisy Images
Enhanced & denoised images
Denoising Module
SSDA 2
Trained with synthetic noisy images only
Contrast Enhancement Module
SSDA 1
Trained with synthetic dark images only
Synthetic Dark & Noisy Images

(a) Module Structure

(b) LLNet

(c) S-LLNet

Figure 1: Architecture of the proposed framework: (a) An autoencoder module is comprised of multiple layers of hidden units, where the encoder is trained by unsupervised learning, the decoder weights are transposed from the encoder and subsequently fine-tuned by error back-propagation; (b) LLNet with a simultaneous contrast-enhancement and denoising module; (c) S-LLNet with sequential contrast-enhancement and denoising modules. The purpose of denoising is to remove noise artifacts often accompanying contrast enhancement.

be highlighted is that the network is trained using images obtained from internet databases that are subsequently synthetically processed (i.e. darkening nonlinearly and adding Gaussian noise) to simulate low-light conditions, since collection of a large number of natural low-light images (sufficient for deep network training) and their well-lit counterparts can be unrealistic for practical use. Despite the fact that LLNet is trained on synthetic images, both synthetic and natural images are used to evaluate the network's performance in denoising and contrast-enhancement.

Aside from the regular LLNet where the network is trained with both darkened and noisy images, we also propose the *staged* LLNet (S-LLNet) which consists of separate modules arranged in series for contrast enhancement (stage 1) and denoising (stage 2). The key distinction over the regular LLNet is that the modules are trained separately with darkened-only training sets and noisy-only training sets. Both structures are presented in Fig. 1. Note, while the S-LLNet architecture provides a greater flexibility of training, it increases the inference time slightly which may be a concern for certain real-time applications. However, customized hardware-acceleration can resolve such issues significantly.

3.2. Network parameters

LLNet is comprised of 3 DA layers, with the first DA layer taking the input image of dimensions 17×17 pixels (i.e. 289 input units). The first DA layer has 2,000 hidden units, the second has 1,600 hidden units, and the third has 1,200 hidden units which becomes the bottleneck layer. Beyond the third DA layer forms the decoding counterparts of the first three layers, thus having 1,600 and 2,000 hidden units for the fourth and fifth layers respectively. Output units have the same dimension as the input, i.e. 289. The network is pre-trained for 30 epochs with pre-training learning rates of 0.1 for the first two DA layers and 0.01 for the last DA layer, whereas finetuning was performed with a learning rate of 0.1 for the first 200 finetuning epochs, 0.01 afterwards, and stops only if the improvement in validation error is less than 0.5%. For the case of S-LLNet, the parameters of each module are identical.

3.3. Training data generation

Training was performed using 422,500 patches, extracted from 169 standard test images[1]. Consistent with current practices, the only pre-processing done was to normalize the image pixels to between zero and one. During the generation of the patches, we produced 2,500 patches from random locations (and with random darkening and noise parameters) from the same image. Note that the images used for generating patches for the training set and the validation set are disjoint in order to reduce the correlation between the training and validation set. By doing so, we avoid correlation between the two sets which has the potential to overestimate the model performance. The 17×17 pixel patches are then darkened nonlinearly using the MATLAB command `imadjust` to randomly apply a gamma adjustment. Gamma correction is a simple but general case with application of a power law formula to images for pixel-wise enhancement with the following expression:

$$I_{\text{out}} = A \times I_{\text{in}}^{\gamma} \tag{1}$$

[1] Dataset URL: http://decsai.ugr.es/cvg/dbimagenes/

where A is a constant determined by the maximum pixel intensity in the image. Intuitively, image is brightened when $\gamma < 1$ while $\gamma = 1$ leaves it unaffected. Therefore, when $\gamma > 1$, the mapping is weighted toward lower (darker) grayscale pixel intensity values.

3.4. Simulating darkness:

A uniform distribution of $\gamma \sim$ Uniform $(2, 5)$ with random variable γ is selected to result in training patches that are darkened to a varying degree. To simulate low quality cameras used to capture images, these original training patches are corrupted by Gaussian noise via the MATLAB function `imnoise` with standard deviation of $\sigma = \sqrt{B(25/255)^2}$, where $B \sim$ Uniform $(0, 1)$. Hence, the final corrupted image and the original image exhibit the following relationship:

$$I_{\text{train}} = n\left(g(I_{\text{original}})\right) \tag{2}$$

where function $g(\cdot)$ represents the gamma adjustment function and $n(\cdot)$ represents the noise function.

Random gamma darkening with random noise levels result in a variety of training images that can increase the robustness of the model. In reality, natural low-light images may also include quantization and Poisson noise (e.g. images captured with imaging sensors such as CCD and CMOS) in addition to Gaussian noise. We chose to focus on the Gaussian-only model for the ease of analysis and as a preliminary feasibility study of the framework trained on synthetic images and applied to natural images. Furthermore, since Gaussian noise is a very familiar yet popular noise model for many image denoising tasks, we can acquire a sense of how well LLNet performs with respect to other image enhancement algorithms. The training set is divided into 211,250 training examples, 211,250 validation samples, and the samples are subsequently randomly shuffled. The training step involves learning the invariant representation of low light and noise with the autoencoder described in section 3.2. While training the model, the network attempts to remove the noise and simultaneously enhance the contrast of these darkened patches. The reconstructed image is compared against the clean version (i.e. bright, noiseless image) by computing the mean-squared error.

When training both LLNet and S-LLNet, each DA is trained by error backpropagation to minimize the sparsity regularized reconstruction loss as described in Xie et al. [25]:

$$\mathcal{L}_{\text{DA}}(\mathcal{D};\theta) = \frac{1}{N}\sum_{i=1}^{N}\frac{1}{2}||y_i - \hat{y}(x_i)||_2^2 \\ + \beta\sum_{j=1}^{K}\text{KL}(\hat{\rho}_j||\rho) + \frac{\lambda}{2}(||W||_F^2 + ||W'||_F^2) \quad (3)$$

where N is the number of patches, $\theta = \{W, b, W', b'\}$ are the parameters of the model, $\text{KL}(\hat{\rho}_j||\rho)$ is the Kullback-Leibler divergence between ρ (target activation) and $\hat{\rho}_j$ (empirical average activation of the j-th hidden unit) which induces sparsity in the hidden layers:

$$\text{KL}(\hat{\rho}_j||\rho) = \rho\log\frac{\rho}{\hat{\rho}_j} + (1-\rho)\log\frac{1-\rho}{1-\hat{\rho}_j} \quad (4)$$

where

$$\hat{\rho}_j = \frac{1}{N}\sum_{i=1}^{N}h_j(x_i) \quad (5)$$

and λ, β and ρ are scalar hyper-parameters determined by cross-validation.

After the weights of the decoder have been initialized, the entire pre-trained network is finetuned using an error back-propagation algorithm to minimize the loss function given by:

$$\mathcal{L}_{\text{SSDA}}(\mathcal{D};\theta) = \frac{1}{N}\sum_{i=1}^{N}||y_i - \hat{y}(x_i)||_2^2 + \frac{\lambda}{L}\sum_{l=1}^{2L}||W^{(l)}||_F^2$$

where L is the number of stacked DAs and $W^{(l)}$ denotes weights for the l-th layer in the stacked deep network. The sparsity inducing term is not needed for this step because the sparsity was already incorporated in the pre-trained DAs.

3.5. Image reconstruction

During inference, the test image is first broken up into overlapping 17×17 patches with stride size of 3×3. The collection of patches is then passed through LLNet to obtain corresponding denoised, contrast-enhanced patches. The patches are averaged and re-arranged back into its original dimensions. From our experiments, we find that using a patching stride of 2×2 or even 1×1 (fully overlapped patches) do not produce significantly superior results. Additionally, increasing the number of DA layers improves the nonlinear modeling capacity of the network. However, a larger model is more computationally expensive to train and we determined that the current network structure is adequate for the present study.

4. Evaluation metrics and compared methods

In this section we present brief descriptions of other contrast-enhancement methods along with the performance metric used to evaluate the proposed framework's performance.

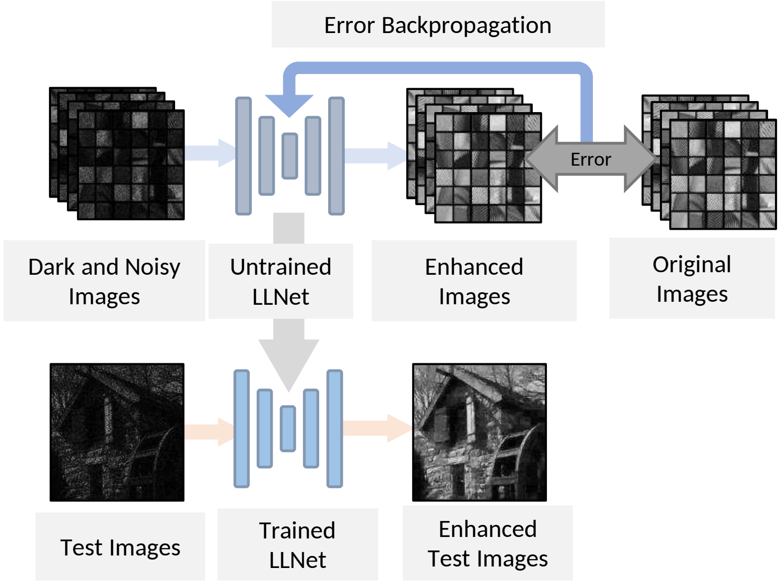


Figure 2: Training the LLNet: Training images are synthetically darkened and added with noise. These images are fed through LLNet where the reconstructed images are compared with the uncorrupted images to compute the error, which is then back-propagated to finetune and optimize the model weights and biases.

4.1. Performance metric

Two metrics are used, namely the peak signal-to-noise ratio (PSNR) and the structural similarity index (SSIM).

4.1.1. Peak signal-to-noise ratio (PSNR)

PSNR quantifies the extent of corruption of original image with noise as well as approximating human perception of the image. It has also been established to demonstrate direct relationship with compression-introduced noise [26]. Roughly, the higher the PSNR, the better the denoised image especially with the same compression code. Basically, it is a modification of the mean squared error between the original image and the reconstructed image. Given a noise-free $m \times n$ monochrome image I and its reconstructed version K, MSE is expressed as:

$$\text{MSE} = \frac{1}{m\,n}\sum_{i=0}^{m-1}\sum_{j=0}^{n-1}[I(i,j) - K(i,j)]^2 \tag{6}$$

The PSNR, in decibels (dB) is defined as:

$$\text{PSNR} = 10 \cdot \log_{10}\left(\frac{\text{max}(I)^2}{\text{MSE}}\right) \tag{7}$$

Here, max(I) is the maximum possible pixel value of the image I.

4.1.2. Structural similarity index (SSIM)

SSIM is a metric for capturing the perceived quality of digital images and videos [6, 27]. It is used to measure the similarity between two images. SSIM quantifies the measurement or prediction of image quality with respect to initial uncompressed or distortion-free image as reference. As PSNR and MSE are known to quantify the absolute error between the result and the reference image, such metrics may not really quantify complete similarity. On the other hand, SSIM explores the change in image structure and being a perception-type model, it incorporates pixel inter-dependencies as well as masking of contrast and pixel intensities. SSIM is expressed as:

$$\text{SSIM}(x,y) = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)} \tag{8}$$

where μ_x is the average of window x, μ_y is the average of window y, σ_x^2 is the variance of x, σ_y^2 is the variance of y, σ_{xy}^2 is the covariance of x and y, $c_1 = (k_1L)^2$ and $c_2 = (k_2L)^2$ are two variables to stabilize the division with weak denominator with $k_1 = 0.01$ and $k_2 = 0.03$ by default, and L is the dynamic range of pixel values.

4.2. Compared methods

This subsection describes several low-light image enhancement methods used for comparison. While we acknowledge other recent non-DL methods [22, 7, 23, 8], the lack of publicly available source codes prevented us from performing detailed comparison.

4.2.1. Histogram equalization (HE)

The histogram of an image is a graphical representation of the intensity distribution of the image which quantifies the number of pixels for each of the intensity values ranging from 0 to 255 when represented with an 8-bit integer. It is a method that improves the contrast of an image by stretching out the intensity range [28, 29, 9]. It maps the original histogram to another distribution with a wider and more uniform distribution (i.e. flatter) so that the intensity values are spread over the entire range. This method is useful in images with backgrounds and foregrounds that are either both bright or both dark, but may not be suitable for images with high dynamic range. In particular, the method can lead to better views of bone structure in x-ray images, and to better detail in photographs that are over or under-exposed.

4.2.2. Contrast-limiting adaptive histogram equalization (CLAHE)

Contrast-limiting adaptive histogram equalization differs from ordinary adaptive histogram equalization in its contrast limiting. In the case of CLAHE, the contrast limiting procedure has to be applied for each neighborhood from which a transformation function is derived [10], as opposed to regular histogram equalization which is carried out in a global manner. CLAHE was developed to prevent the over-amplification of noise that arise in adaptive histogram equalization.

4.2.3. Gamma adjustment (GA)

The simple form of gamma correction is outlined in Eqn. (2). Gamma curves illustrated with $\gamma > 1$ have exactly the opposite effect as those generated with $\gamma < 1$. It is important to note that gamma correction reduces toward the identity curve when $\gamma = 1$. In other words, any image corrected with $\gamma = 1$ results in the exact same image. As discussed earlier in section 3.3, the image is generally brightened when $\gamma < 1$ and darkened when $\gamma > 1$.

4.2.4. Histogram equalization with 3D block matching (HE+BM3D)

BM3D is the current state-of-the-art algorithm for image noise removal presented by [15]. It uses a collaborative form of Wiener filter for high dimensional block of patches by grouping similar 2D blocks into a 3D data array, and then denoising the grouped patches jointly. The denoised patches from the stack are applied back on the original images by a voting mechanism which removes noise from the considered region.

In this work, we decided to first equalize the contrast of the test image, then use BM3D as a denoiser to remove the noise resulting from histogram equalization. Previously, we also attempted to reverse the order, i.e. use BM3D to remove noise from the low-light images first and followed by contrast enhancement. Since BM3D removes noise by applying denoised patches, the blob-shaped patch boundaries are significantly amplified and become extremely pronounced when histogram equalization is applied. This produces non-competitive results which make comparison unfair. Hence, we ensure that BM3D is performed after histogram equalization when reporting the results.

5. Results and discussion

In this section, we evaluate the performance of our framework against the methods outlined above on standard images shown in Fig. 3. Test images are darkened with $\gamma = 3$, where noisy versions contain Gaussian noise of $\sigma = 18$ and $\sigma = 25$, which are typical values for image noise under poor illumination and/or high temperature; these parameters correspond to scaled variances of $\sigma_s^2 = 0.005$ and $\sigma_s^2 = 0.010$ respectively if the pixel intensities are in 8-bit integers ($\sigma_s = \sigma/255$ where $\sigma_s \in [0, 1]$ and $\sigma \in [0, 255]$). These parameters are first fixed in order to study the effectiveness of each method in contrast enhancement and denoising. For a more generalized set of synthetic test images, darkening and noise addition are performed using randomized values of $\gamma \in [1, 4]$ and $\sigma \in [0, 25]$.

Histogram equalization is performed by using the MATLAB function `histeq`, whereas CLAHE is performed with the function `adapthisteq` with default parameters (8×8 image tiles, contrast enhancement limit of 0.01, full range output, 256 bins for building contrast enhancing transformation, uniform histogram distribution, and distribution parameter of 0.4). Gamma adjustment is performed on dark images with $\gamma = 1/3$ unless otherwise stated. For the hybrid 'HE+BM3D' method, we first applied histogram equalization to enhance image contrast before using the BM3D code developed by Dabov et al. [15] as a denoiser, where the noise standard deviation input parameter for BM3D is set to $\sigma = 25$ (the highest noise level of the test image). Both LLNet and S-LLNet outputs are reconstructed with overlapping 17×17 patches of stride size 3×3. Training was performed on

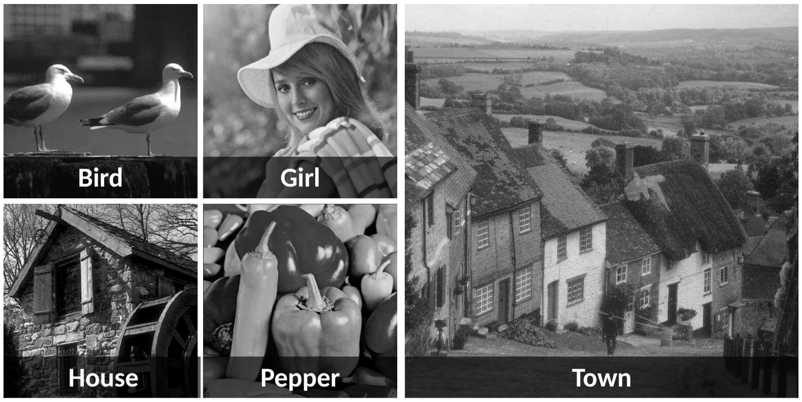


Figure 3: Original standard test images used to compute PSNR.

NVIDIA's TITAN X GPU using Theano's deep learning framework [30, 31] and took approximately 30 hours. Enhancing an image with dimension of 512×512 pixels took 0.42 s on GPU.

5.1. Algorithm adaptivity

Ideally, an already-bright image should no longer be brightened any further. To test this, different enhancement algorithms are performed on a normal, non-dark and noiseless image. Fig. 4A shows the result when running the 'Town' image through various algorithms. LLNet outputs a slightly brighter image, but not to the degree that everything appears over-brightened and washed-out like the output of GA if GA is blindly applied with $\gamma = 1/3$. This shows that in the process of learning low-light features, LLNet successfully learns the necessary degree of required brightening that should be applied to the image. However, when evaluating contrast enhancement via visual inspection, histogram equalization methods (i.e. HE, CLAHE, HE+BM3D) provide superior enhancement given the original image. When tested with other images (namely, 'Bird', 'Girl', 'House', 'Pepper', etc.) as shown in Table 1, HE-based methods generally fared slightly better with higher PSNR and SSIM.

5.2. Enhancing artificially darkened images

Fig. 4B shows output of various methods when enhancement is applied to a 'Town' image darkened with $\gamma = 3$. Here, LLNet achieves the highest PSNR followed by GA, but the the other way round when evaluated with SSIM. The high similarity between the GA-enhanced image with the original is expected because

Table 1: PSNR and SSIM of outputs using different enhancement methods. 'Bird' means the non-dark and noiseless (i.e. original) image of Bird. 'Bird-D' indicates a darkened version of the same image. 'Bird-D+GN18' denotes a darkened Bird image with added Gaussian noise of $\sigma = 18$, whereas 'Bird-D+GN25' denotes darkened Bird image with added Gaussian noise of $\sigma = 25$. Bolded numbers corresponds to the method with the highest PSNR or SSIM. Asterisk (*) denotes our framework.

PSNR(dB)/SSIM	**Dark**	**HE**	**CLAHE**	**GA**	**HE+BM3D**	**LLNet***	**S-LLNet***
Bird	N/A	11.22 / 0.63	**21.55 / 0.90**	8.93 / 0.66	11.27 / 0.69	17.61 / 0.84	18.35 / 0.85
Bird-D	12.27 / 0.18	11.28 / 0.62	15.15 / 0.52	**29.53 / 0.86**	11.35 / 0.71	20.09 / 0.69	15.87 / 0.52
Bird-D+GN18	12.56 / 0.14	9.25 / 0.09	14.63 / 0.11	14.10 / 0.11	9.98 / 0.13	**20.17 / 0.66**	18.59 / 0.54
Bird-D+GN25	12.70 / 0.12	9.04 / 0.08	13.60 / 0.09	13.07 / 0.08	9.72 / 0.11	21.87 / **0.64**	**22.53** / 0.63
Girl	N/A	18.24 / 0.80	17.02 / 0.70	11.08 / **0.81**	**18.23** / 0.69	18.17 / 0.77	14.31 / 0.72
Girl-D	9.50 / 0.50	18.27 / 0.80	14.36 / 0.66	**47.32 / 1.00**	18.26 / 0.69	23.61 / 0.76	21.21 / 0.72
Girl-D+GN18	9.43 / 0.21	16.07 / 0.26	12.95 / 0.17	17.21 / 0.29	19.28 / 0.53	19.93 / **0.66**	**21.97** / 0.64
Girl-D+GN25	9.39 / 0.15	15.33 / 0.19	12.09 / 0.12	15.37 / 0.20	18.50 / 0.39	20.08 / **0.60**	**22.60** / 0.59
House	N/A	13.36 / 0.70	**18.89 / 0.81**	10.21 / 0.59	13.24 / 0.61	11.35 / 0.55	10.52 / 0.46
House-D	12.12 / 0.33	12.03 / 0.65	16.81 / 0.60	**28.79 / 0.83**	11.92 / 0.54	21.80 / 0.64	18.31 / 0.46
House-D+GN18	12.19 / 0.29	10.55 / 0.33	15.48 / 0.35	14.44 / 0.34	11.39 / 0.42	**21.01 / 0.57**	19.31 / 0.47
House-D+GN25	12.16 / 0.26	10.09 / 0.29	14.08 / 0.29	13.26 / 0.29	10.94 / 0.37	**20.68 / 0.54**	19.84 / 0.47
Pepper	N/A	18.61 / 0.90	18.27 / 0.76	10.29 / 0.72	**18.61 / 0.84**	10.53 / 0.66	10.01 / 0.64
Pepper-D	10.45 / 0.37	18.45 / 0.85	15.46 / 0.61	**32.97 / 0.90**	18.45 / 0.80	21.52 / 0.79	19.27 / 0.70
Pepper-D+GN18	10.45 / 0.19	14.69 / 0.21	14.49 / 0.17	15.74 / 0.22	16.97 / 0.57	**22.76 / 0.68**	22.07 / 0.64
Pepper-D+GN25	10.41 / 0.15	13.67 / 0.15	13.31 / 0.13	14.33 / 0.16	15.96 / 0.36	22.94 / 0.61	**23.17 / 0.61**
Town	N/A	17.55 / 0.79	16.35 / 0.69	10.02 / 0.76	**17.70** / 0.76	16.28 / **0.80**	16.03 / 0.78
Town-D	10.17 / 0.36	17.55 / 0.79	15.00 / 0.65	**36.80 / 0.97**	17.72 / 0.76	21.42 / 0.75	19.90 / 0.68
Town-D+GN18	10.19 / 0.18	14.85 / 0.25	13.34 / 0.17	15.53 / 0.24	17.51 / 0.41	19.85 / **0.64**	**20.52** / 0.59
Town-D+GN25	10.21 / 0.14	14.22 / 0.20	12.40 / 0.13	14.08 / 0.17	16.62 / 0.32	21.63 / **0.60**	**22.89** / 0.58
Average	10.95 / 0.24	14.22 / 0.48	15.26 / 0.43	18.65 / 0.51	15.18 / 0.54	**19.66 / 0.67**	18.86 / 0.61

Table 2: Average PSNR and SSIM over 90 synthetic and 6 natural test images. Synthetic test images are randomly darkened with $\gamma \in [1, 4]$ and Gaussian noise levels of $\sigma \in [0, 25]$. Natural test images are taken under natural low-light conditions. Because gamma darkening is performed randomly for this set of images, we search for the optimal γ parameter that results in the highest SSIM ($\gamma = 0.05 : 0.05 : 1$) when applying gamma adjustment. Note that searching for the optimal parameter is infeasible in reality because no reference image is available. The number reported within the parentheses is the number of winning instances among 90 synthetic test images and 6 natural test images. Asterisk (*) denotes our framework.

Test Items	**Dark**	**HE**	**CLAHE**	**GA**	**HE+BM3D**	**LLNet***	**S-LLNet***
Average PSNR (dB), synthetic	15.7902	13.7765 (0)	14.3198 (5)	15.2692 (6)	15.1653 (20)	**19.8109 (52)**	18.2248 (7)
Average SSIM, synthetic	0.4111	0.3524 (3)	0.3255 (1)	0.4345 (2)	0.5127 (17)	**0.6912 (65)**	0.6066 (2)
Average PSNR (dB), natural	8.2117	11.7194 (0)	9.9473 (0)	14.6664 (2)	11.9596 (1)	**15.1154 (2)**	14.4851 (1)
Average SSIM, natural	0.1616	0.2947 (0)	0.3611 (0)	0.5338 (0)	0.5437 (2)	**0.6152 (3)**	0.5467 (1)

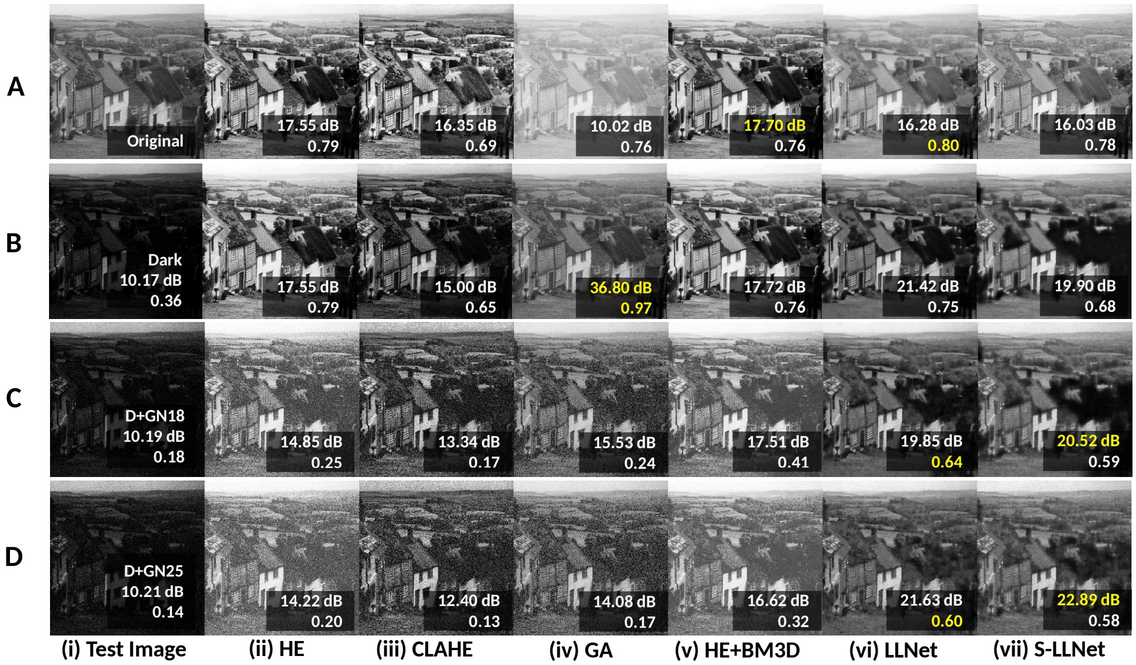


Figure 4: Comparison of methods of enhancing ‘Town’ when applied to (A) original already-bright, (B) darkened, (C) darkened and noisy ($\sigma = 18$), and (D) darkened and noisy ($\sigma = 25$) images. Darkening is done with $\gamma = 3$. The numbers with units dB are PSNR, the numbers without are SSIM. Best viewed on screen.

the optimal gamma readjustment parameter essentially reverses the process close to the original intensity levels. In fact, when tested with other images, the highest scores for darkened-only images are achieved by only one of LLNet, S-LLNet or GA where HE, CLAHE, and HE+BM3D fail. Results tabulated in Table 1 highlight the advantages and broad applicability of the deep autoencoder approach with LLNet and S-LLNet.

5.3. Enhancing darkened images in the presence of synthetic noise

To simulate dark images taken with regular or subpar camera sensors, Gaussian noise is added to the synthetic dark images. Fig. 4C and 4D presents a gamma-darkened ‘Town’ image corrupted with Gaussian noise of $\sigma = 18$ and $\sigma = 25$, respectively. For these test images, both LLNet and S-LLNet attained superior PSNR and SSIM over other methods, as shown in Table 1. Histogram

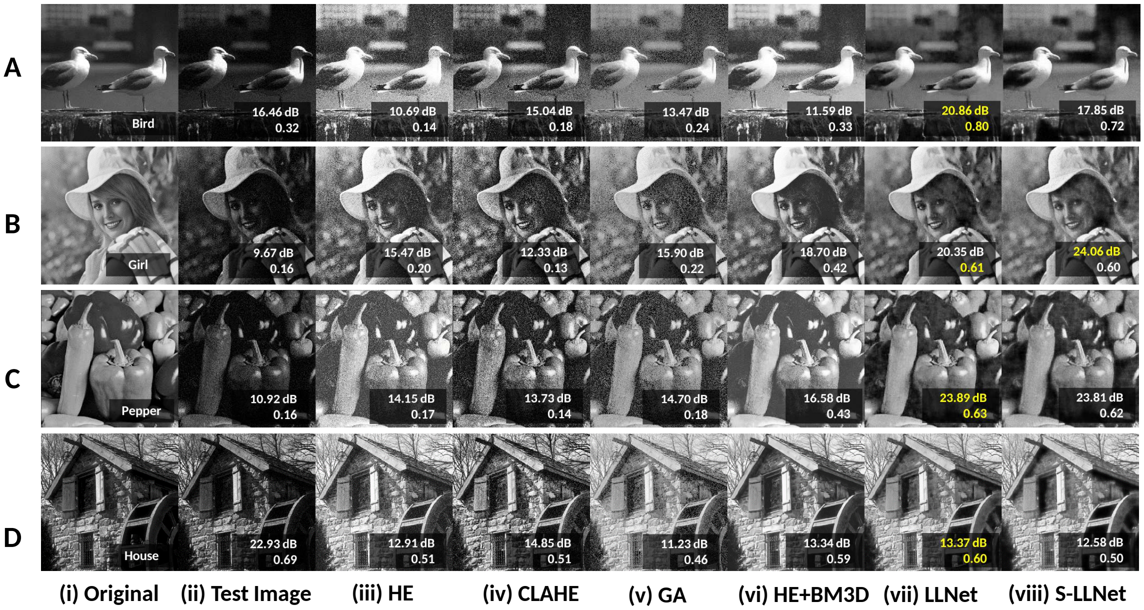


Figure 5: Comparison of methods on randomly darkened/noise-added synthetic test images of (A) Bird, (B) Girl, (C) Pepper, and (D) House. Darkening and noise addition are done using randomized values of $\gamma \in [1, 4]$ and $\sigma \in [0, 25]$. The numbers with units dB are PSNR, the numbers without are SSIM. Best viewed on screen.

equalization methods fail due to the intensity of noisy pixels being equalized and produced detrimental effects to the output images. Additionally, BM3D is not able to effectively denoise the equalized images with parameter $\sigma = 25$ since the structure of the noise changes during the equalization process.

Instead of using fixed values of γ and σ for darkening and noise addition, we generated 90 images using randomized values of $\gamma \in [1, 4]$ and $\sigma \in [0, 25]$. Next, the performance of each algorithm is evaluated on these 90 images and the average PSNR and SSIM are computed and tabulated in Table 2. Four out of the 90 results are shown in Fig. 5. In the table, both the average SSIM and PSNR of the standalone LLNet achieved the best performance compared to other methods, and generally fares better than S-LLNet. It appears that S-LLNet only produces the best enhancement at very dark and high noise levels. However, when the γ and σ parameters vary at slightly lower levels, then LLNet outperforms S-LLNet. This

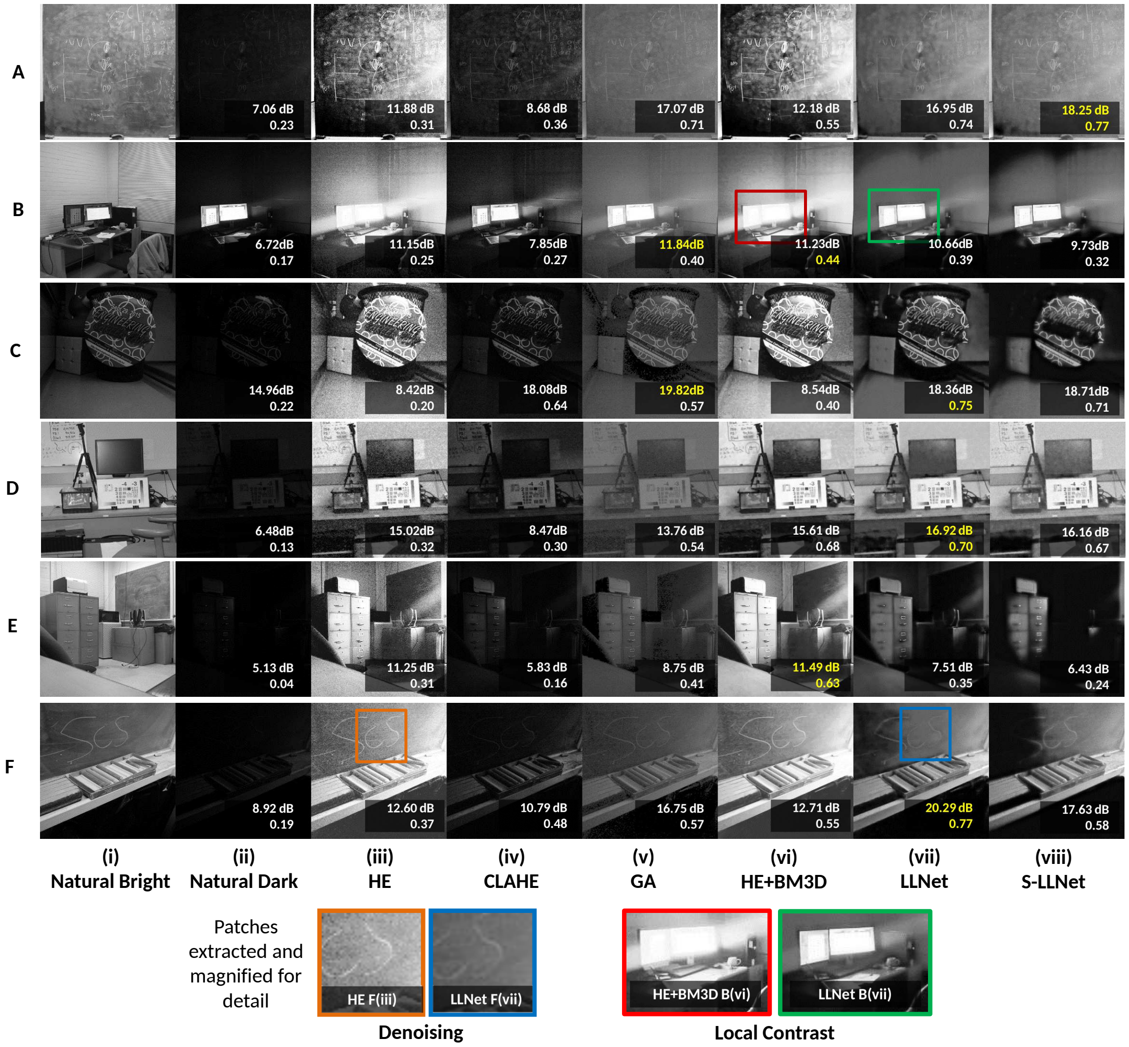


Figure 6: Comparison of methods of enhancing naturally dark images of (A) chalkboard, (B) computer, (C) objects, (D) chart, (E) cabinet, and (F) writings. Selected regions are enlarged to demonstrate the denoising and local contrast enhancement capabilities of LLNet. HE (including HE+BM3D) results in overamplification of the light from the computer display whereas LLNet was able to avoid this issue.

is because LLNet performs both contrast enhancement and denoising simultaneously, rather than doing the tasks in a stage-wise manner which implicitly assumes independence between the two tasks.

5.4. Application on natural low-light images

When working with downloaded images, a clean reference image is available for computing PSNR and SSIM. However, reference images may not be available in real life when working with naturally dark images. Since this is a controlled experiment, we circumvented the issue by mounting an ordinary cell-phone (Nexus 4) camera on a tripod to capture pictures in an indoor environment with both lights on and lights off. The picture with lights on are used as the reference images for PSNR and SSIM computations, whereas the picture with lights off becomes the natural low-light test image. Although the bright pictures cannot be considered as the ground truth, it provides a reference point to evaluate the performance of various algorithms. Performance of each enhancement method is shown in Fig. 6. While histogram equalization greatly improves the contrast of the image, it corrupts the output with a large noise content. In addition, the method suffers from over-amplification in regions where there is a very high intensity brightness in dark regions, as shown by blooming effect on the computer display in panel 6B(vi) and 6B(vii). CLAHE is able to improve the contrast without significant blooming of the display, but like HE it tends to amplify noise within the images. LLNet performs significantly well with its capability to suppress noise in most of the images while improving local contrast, as shown in the magnified patches at the bottom of Fig. 6.

5.5. Training with Gaussian vs. Poisson noise

In certain natural low-light scenarios, the underlying noise profile can be properly modeled by photon shot noise or Poisson noise which is a type of electronic noise. The dominant noise in darker regions of an image from an image sensor is usually caused by statistical quantum fluctuations, that is, the variation in the number of photons sensed at a given exposure level. From a mathematical perspective, while Gaussian noise is typically generated separately and added independently to each individual pixels from the original image, Poisson noise takes the original pixel intensities into account and generates new intensities from a Poisson process. In other words, Gaussian noise is independent of the original intensities in the image but Poisson noise is correlated with the intensity of each pixel. A visual example is provided in Fig. 7 to show the differences between Gaussian noise and Poisson noise at different light intensities (i.e. photon count). Since most imaging sensors such as CCD and CMOS suffers from Poisson noise when capturing low-light images, training the model with synthetic images with Poisson noise has potential advantages in enhancing natural low-light images. An experimental comparison between two training schemes–with Poisson vs. with Gaussian noise–is presented in Fig. 8.

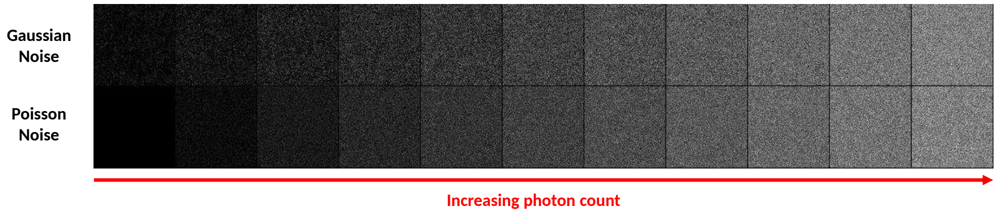


Figure 7: The result of *adding* Gaussian noise and *applying* Poisson noise with increasing photon count (normalized from 0 to 1 with a step size of 0.1). Although the noise levels between the two noise types look similar at higher photon count, the first three columns look very different. Best viewed on screen.

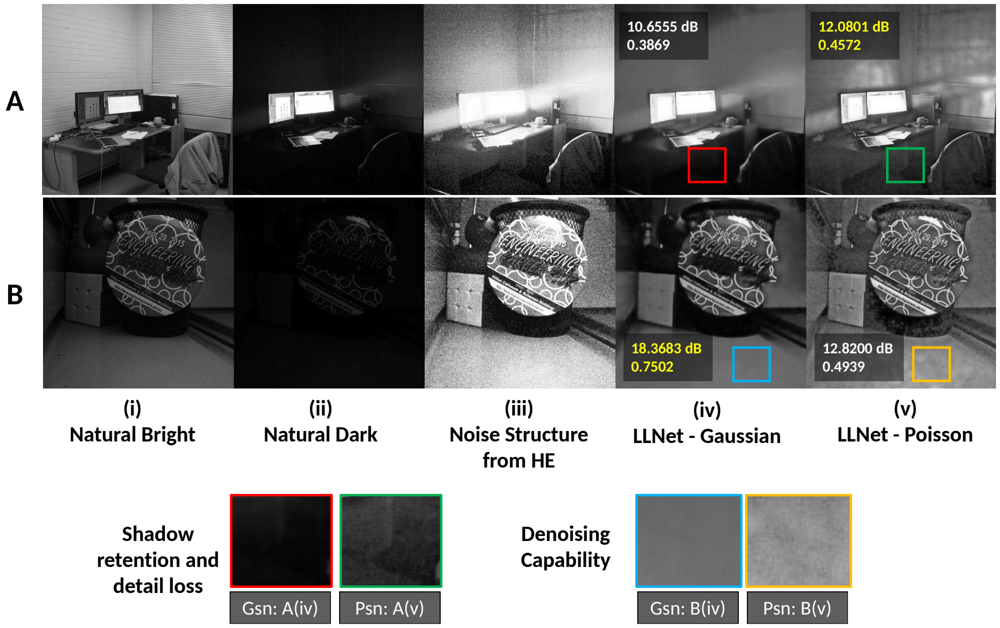


Figure 8: Natural low-light image enhancement results of LLNet trained with Gaussian noise (LLNet-G) and Poisson noise (LLNet-P). 'Gsn' and 'Psn' are abbreviations for Gaussian and Poisson respectively. Best viewed on screen.

In Fig. 8, the outputs of LLNet trained with Gaussian noise (*LLNet-G*) generally appear smoother but suffers some loss in detail due to the retention of shadows. On the other hand, the model trained with Poisson noise (*LLNet-P*) produces comparatively noisier images but with sharper details. The reason contributing to this disparity lies in the nature of the training set.

To explain why LLNet-G tends to retain the shadows but denoises better than LLNet-P, recall that from Fig. 7, darker training patches are less affected by Poisson noise compared to Gaussian noise. Therefore, LLNet-G is able to see more noisy training examples (specifically noisy dark patches) and learn how to denoise them. Furthermore, when a very dark patch is corrupted with Gaussian noise, pixel intensities that become negative are clipped to 0. Hence, this raises the average pixel intensity of a single dark training patch and causes this particular patch appear more gray than black. During patch-wise enhancement, LLNet-G may encounter a gray patch and mistake it for a noisy dark patch. Ultimately, the gray patches are darkened and consequently contribute to dark shadows being retained in the enhanced image. On the contrary, dark patches used to train LLNet-P are least affected by Poisson noise, which in turn reduces the number of noisy examples where LLNet-P learns the denoising function from. The resultant effect is a lower denoising capability of LLNet-P compared to LLNet-G, but with the gained advantage where shadows are also enhanced to bring out relevant details.

Note, with sufficient training data and optimized hyperparameters, a deep autoencoder can learn to approximate almost any nonlinear denoising function. On that account, a union of the two training sets (i.e. with Gaussian and Poisson noise) can be used to train a new LLNet where both noise types are taken into consideration. While there are certainly many other ways to further improve the performance of LLNet (e.g. hyperparameter optimization, ensemble methods, and more rigorous process modeling), we show that the notion of *transfer learning* can be realized with appropriate training data generation schemes that adequately model a real world process. Thus a model trained with synthetic images can indeed be applied to enhance natural low-light images with competitive performance.

5.6. Denoising capability, image sharpness, and patch size

There is a trade-off between denoising capability and the perceived sharpness of the enhanced image. While higher PSNR indicates a higher denoising capability, this metric favors smoother edges. Therefore, images that are less sharp often achieve a higher PSNR. Hence, SSIM is used as a complementary metric to evaluate the gain or loss in perceived structural information. From the experiments, a relationship between denoising capability (PSNR), similarity levels (SSIM) and image sharpness is found to be dependent on the dimensions of the denoised patch relative to the test image. A smaller patch size implies finer-grain enhancement over the test image, whereas a larger patch size implies coarser enhancement. Because natural images may also come in varying heights and widths, the relative patch size–a dimensionless quantity that relates the patch size to the dimensions of the test image, r–is defined as:

$$r = \frac{d_p}{d_i} = \frac{\sqrt{w_p^2 + h_p^2}}{\sqrt{w_i^2 + h_i^2}}$$

where quantities d, w, and h denote the diagonal length, width, and height in pixels, with subscripts p and i referring to the patch and test image respectively. Relative patch size may also be thought as the size of the receptive field on a test image. From the results, it is observed that when the relative patch size decreases, object edges appear sharper at the cost of having more noise. However, there exists an optimal patch size resulting in an enhanced image with the highest PSNR or SSIM (as shown in Fig. 9 and Fig. 10.). If the optimal patch size is selected based on PSNR, the resulting image will have the lowest noise levels but is less sharp. If the smallest patch size is selected, then the resulting image has the highest sharpness where more details can be observed but with the expense of having more noise. Choosing the optimal patch size based on SSIM produces a more well-balanced result in terms of denoising capability and image sharpness.

We included a natural test image where the US Air Force (USAF) resolution test chart is shown. The test chart consists of groups of three bars varying in sizes labeled with numbers which conforms to the MIL-STD-150A standard set by the US Air Force in 1951. Originally, this test chart is used to determine the resolving power of optical imaging systems such as microscopes, cameras, and image scanners. For the present study, we used this test chart to visually compare the trade-off denoising capability and image sharpness using different relative patch sizes. The results are shown in Fig. 10.

5.7. Prior knowledge on input

HE can be easily performed on images without any input parameters. Like HE, CLAHE can also be used without any input parameters where the performance can be further finetuned with various other parameters such as tile sizes, contrast output ranges, etc. Gamma adjustment and BM3D both require prior knowledge of the input parameter (values of γ and σ, respectively), thus it is often necessary to finetune the parameters by trial-and-error to achieve the best results. The advantage of using deep learning-based approach, specifically using LLNet

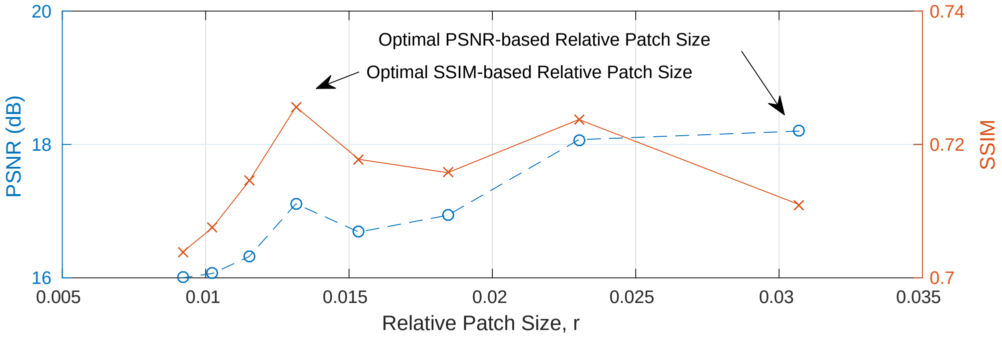
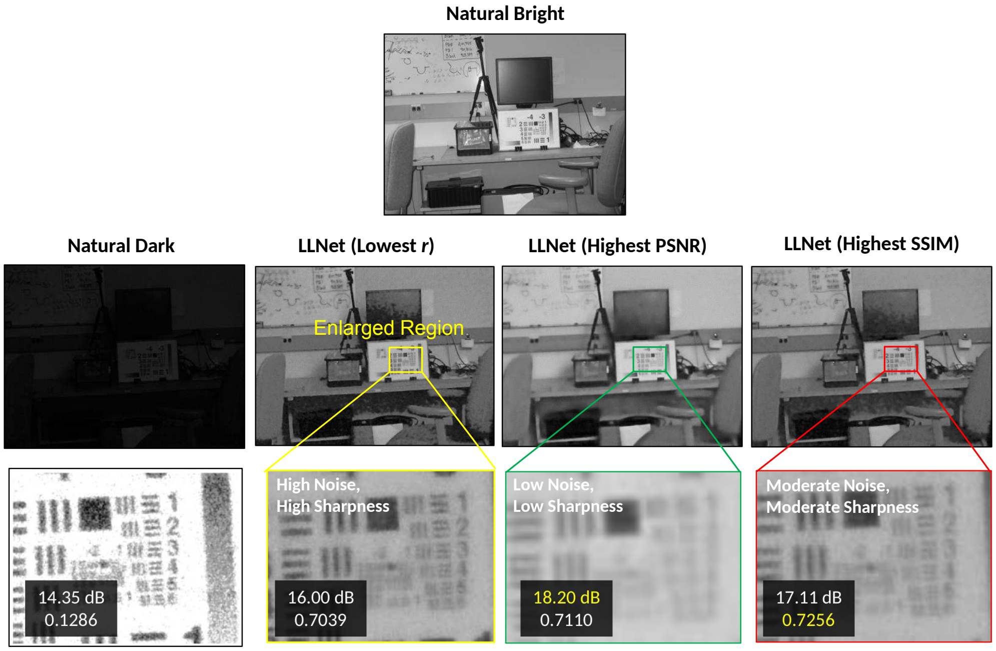


Figure 9: Relative patch size vs PSNR and SSIM. The picture with highest PSNR has the highest denoising capability but least sharp. Picture with lowest r has the least denoising capability but has the highest image sharpness. Picture with the highest SSIM balances between image sharpness and denoising capability.

and S-LLNet, is that after training the model with a large variety of darkened and noisy images with proper choice of hyper-parameters, there is no need for meticulous hand-tuning during testing/practical use. This effectively reduces the burden on the end-user. The model automatically extracts and learns the underlying features from low-light images. Essentially, this study shows that a deep model that has been trained with varying degrees of darkening and noise levels can be used for many real-world problems without detail knowledge of camera and environment.

5.8. Features of low-light images

To gain an understanding on what features are learned by the model, weights linking the input to the first layer of the trained model can be visualized by plotting the values of the weight matrix as pixel intensity values (Fig. 11). In a regular LLNet where both contrast enhancement and denoising are learned simultaneously, the weights contain blob-like structures with prominent coarse-looking textures. Decoupling the learning process (in the case of S-LLNet) allows us to acquire a better insight. We observe that blob-like structures are learned when the model is trained for the task of contrast enhancement. The shape of the features suggest that contrast enhancement considered localized features into account; if a region is dark, then the model brightens it based on the context in the patch (i.e. whether the edge of an object is present or not). On the other hand, feature detectors for the denoising task appears noise-like, albeit in a finer-looking texture compared to the on coarser ones from the integrated LLNet. These features shows that the denoising task is mostly performed in an overall manner. Note that while the



(a) Histogram Equalization as benchmark

(b) **LLNet Output:** Lowest relative patch size

(c) **LLNet Output:** Optimal relative patch size based on PSNR

(d) **LLNet Output:** Optimal relative patch size based on SSIM

Figure 10: Evaluation on US Air Force (USAF) resolution test chart. There exist optimal relative patch sizes that result in the highest PSNR or SSIM after image enhancement (using LLNet). Note that the result enhanced with histogram equalization is shown to highlight the loss in detail of the natural dark image (where the main light is turned off) compared to the natural bright image.

visualizations presented in [21] show prominent Gabor-like features at different orientations for the denoising task, the Gabor-like features are not apparent in the present study because the training data consists of multiple noise levels rather than a fixed one. The distinction between feature detectors and feature generators is highlighted in Fig. 12 and a comparison of superior and inferior weights is shown in Fig. 13.

5.9. Hyper-parameters, network architecture, and performance

Table 3 shows the average PSNR and SSIM values evaluated on the set of 90 synthetic dark images enhanced with the trained model of different hyper-parameters and network architecture. As we are interested in the overall image enhancement performance, we used the implementation with the highest SSIM, as opposed to PSNR, for the reported results. From the results, smaller batch sizes

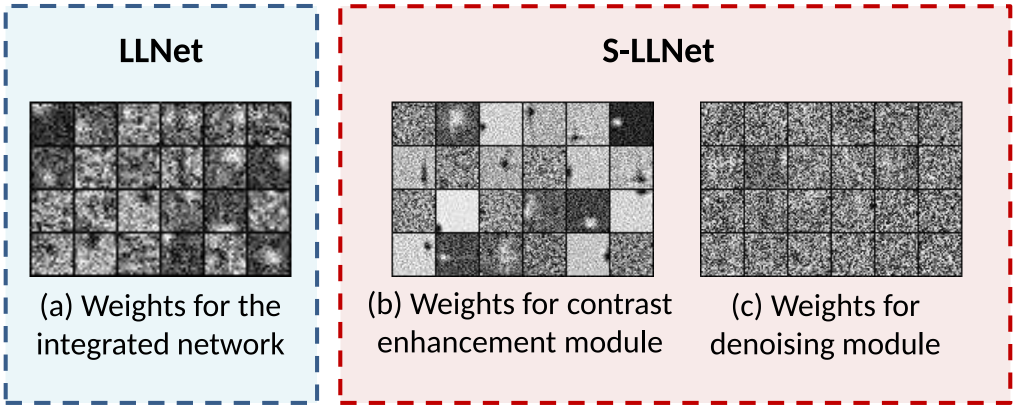

LLNet
S-LLNet
(a) Weights for the integrated network
(b) Weights for contrast enhancement module
(c) Weights for denoising module

Figure 11: Feature detectors can be visualized by plotting the weights connecting the input to the hidden units in the first layer. These weights are selected randomly.

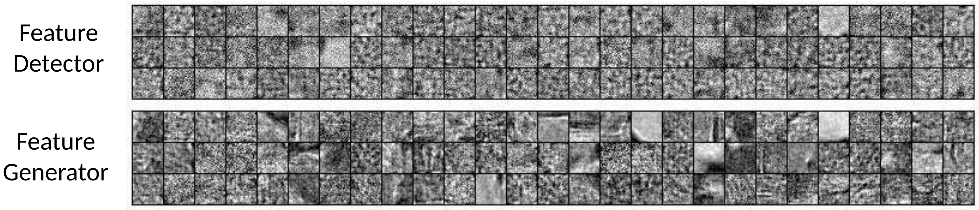


Figure 12: Random selection of weights from the first layer (feature detectors) and weights from the output layer (feature generators) for the integrated LLNet model trained with a patch size of 21×21. Patterns in the output weights are similar to patterns in the first hidden layer weights since tied weights are used.

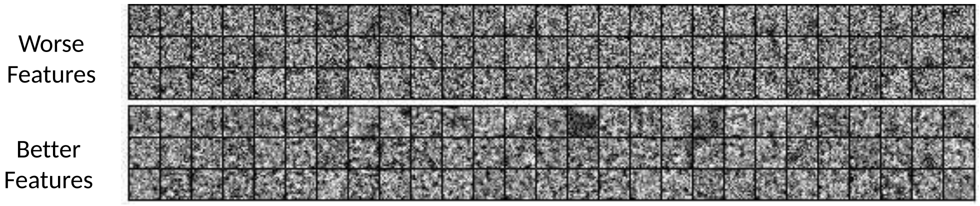


Figure 13: Random selection of first layer weights from an integrated LLNet model trained in batches of 1000 and 50, respectively. The superior model (i.e. batch size 50) learns features that appear more distinctive.

Table 3: Average PSNR evaluated on the set of 90 test images enhanced with trained model of different hyper-parameters and network architecture. The implemented model is marked by an asterisk (*), whereas the PSNR and SSIM for the best model are presented in bolded typeface.

Model Description	Network Architecture (# Hidden Units)	PSNR (dB)	SSIM
*Batch Size 50	2000-1600-1200-1600-2000	19.8109	**0.6912**
Batch Size 500	2000-1600-1200-1600-2000	**20.0979**	0.6710
Batch Size 1000	2000-1600-1200-1600-2000	20.0550	0.6600
Patch Size 13×13	2000-1600-1200-1600-2000	**19.8281**	0.6271
*Patch Size 17×17	2000-1600-1200-1600-2000	19.8109	**0.6912**
Patch Size 21×21	2000-1600-1200-1600-2000	19.5877	0.6375
Patch Size 25×25	2000-1600-1200-1600-2000	19.4637	0.6355
3-layer SDA	1600-1200-1600	**20.2458**	0.6845
*5-layer-SDA	2000-1600-1200-1600-2000	19.8109	**0.6912**
7-layer SDA	2400-2000-1600-1200-1600-2000-2400	19.1717	0.6480
Narrowest	500-400-300-400-500	19.3688	0.6774
Narrow	1000-800-600-800-1000	19.8056	0.6879
*Regular	2000-1600-1200-1600-2000	**19.8109**	**0.6912**
Wide	4000-3200-2400-3200-4000	19.7214	0.6730

result in noisier gradients during the update and may help in escaping local minima during optimization. Hence, we see that the SSIM increases with a sufficiently small batch size. No clear trend is observed in terms of PSNR with varying batch sizes. A patch size of 13×13 resulted in the highest average PSNR whereas a patch size of 17x17 resulted in the highest SSIM. This result has been discussed in earlier sections and is consistent with the findings on the relationship between the relative patch size and PSNR and SSIM, where the selection of the optimal patch size requires considering the trade-off between image sharpness with denoising power. On the other hand, the number of hidden layers must be chosen such that it adequately captures the nonlinearity in the data (i.e. architecture is not too shallow) while avoiding the vanishing gradient problem which inhibits learning (i.e. architecture is not too deep). The same effect is observed for the width of the architecture.

Note that we explored how varying the hyperparameters one at a time affects the model performance. However, if we use all the optimal hyperparameters discovered independently, such model will not necessarily result in a globally optimal performance. Hence, it may be desirable to explore the hyperparameter space randomly [32] as opposed to doing the search in a sequential manner.

6. Conclusions and future works

A variant of the stacked sparse denoising autoencoder was trained to learn the brightening and denoising functions from various synthetic examples as filters which are then applied to enhance naturally low-light and degraded images. Results show that deep learning based approaches are suitable for such tasks for natural low-light images of varying degree of degradation. The proposed LLNet framework compete favorably with currently used image enhancement methods such as histogram equalization, CLAHE, gamma adjustment, and hybrid methods such as applying HE first and subsequently using a state-of-the-art denoiser such as BM3D. While the performance of some of these methods remain competitive in some scenarios, our framework was able to adapt and perform consistently well across a variety of (lighting and noise) situations. This implies that deep autoencoders are effective tools to learn underlying signal characteristics and noise structures from low-light images without hand-crafting. Some envisaged improvements and future research directions are: (i) Training with quantization artifacts to simulate a more realistic situation; (ii) explore other deep architectures for the purpose of natural low-light image enhancement; (iii) include de-blurring capability explicitly to increase sharpness of image details; (iv) train models that are robust and adaptive to a combination of noise types with extension beyond low-light scenarios such as foggy and dusty scenes; (v) perform subjective evaluation by a group of human users.

7. Acknowledgements

This work was supported in part by the Iowa State Regents Innovation Funding and Rockwell Collins Inc. We gratefully acknowledge the support of NVIDIA Corporation with the donation of the GeForce GTX TITAN Black GPU used for this research.

8. References

[1] A. Krizhevsky, I. Sutskever, G. E. Hinton, Imagenet classification with deep convolutional neural networks, NIPS 2012: Neural Information Processing Systems, Lake Tahoe, Nevada.

[2] C. Couprie, C. Farabet, L. Najman, Y. LeCun, Indoor semantic segmentation using depth information, in: ICLR, 2013.

[3] S. Sarkar, V. Venugopalan, K. Reddy, J. Ryde, M. Giering, N. Jaitly, Occlusion edge detection in rgbd frames using deep convolutional neural networks, Proceedings of IEEE High Performance Extreme Computing Conference, (Waltham, MA).

[4] S. Sarkar, K. G. Lore, S. Sarkar, V. Ramanan, S. R. Chakravarthy, S. Phoha, A. Ray, Early detection of combustion instability from hi-speed flame images via deep learning and symbolic time series analysis.

[5] K. G. Lore, N. Sweet, K. Kumar, N. Ahmed, S. Sarkar, Deep value of information estimators for collaborative human-machine information gathering, in: International Conference of Cyber-physical Systems (ICCPS), Vienna, Austria, 2016.

[6] A. Loza, D. Bull, P. Hill, A. Achim, Automatic contrast enhancement of low-light images based on local statistics of wavelet coefficients, Elsevier Digital Signal Processing 23 (6) (2013) 1856–1866.

[7] X. Dong, G. Wang, Y. A. Pang, W. Li, J. G. Wen, W. Meng, Y. Lu, Fast efficient algorithm for enhancement of low lighting video, in: Multimedia and Expo (ICME), 2011 IEEE International Conference on, IEEE, 2011, pp. 1–6.

[8] A. Yamasaki, H. Takauji, S. Kaneko, T. Kanade, H. Ohki, Denighting: Enhancement of nighttime images for a surveillance camera, in: Pattern Recognition, 2008. ICPR 2008. 19th International Conference on, IEEE, 2008, pp. 1–4.

[9] S. M. Pizer, E. P. Amburn, J. D. Austin, R. Cromartie, A. Geselowitz, T. Greer, B. ter Haar Romeny, J. B. Zimmerman, K. Zuiderveld, Adaptive histogram equalization and its variations, Computer vision, graphics, and image processing 39 (3) (1987) 355–368.

[10] E. Pisano, S. Zong, B. Hemminger, M. DeLuce, E. Johnston, K. Muller, P. Braeuning, S. Pizer, Contrast limited adaptive histogram equalization image processing to improve the detection of simulated spiculations in dense mammograms, Journal of Digital Imaging 11 (4) (1998) 193–200.

[11] R. Krutsch, D. Tenorlo, Histogram equalization, Application Note AN4318, Freescale Semiconductors Inc (June 2011).

[12] M. Kaur, J. Kaur, J. Kaur, Survey of contrast enhancement techniques based on histogram equalization, International Journal of Advanced Computer Science and Applications 2 (7) (2011) 137–141.

[13] X. Wu, A linear programming approach for optimal contrast-tone mapping, IEEE Transaction on Image Processing 20 (5) (2011) 1262–1272.

[14] R. Gonzalex, R. Woods, Digital image Processing, 2nd Edition, no. 0-201-28075-8 in 0-201-28075-8, Prentice Hall,, upper saddle Rivers,New Jersey, 2001.

[15] K. Dabov, A. Foi, V. Katkovnik, K. Egiazarian, Image denoising by sparse 3-d transform-domain collaborative filtering, Image Processing, IEEE Transactions on 16 (8) (2007) 2080–2095.

[16] M. Elad, M. Aharon, Image denoising via sparse and redundant representations over learned dictionaries, Image Processing, IEEE Transactions on 15 (12) (2006) 3736–3745.

[17] R. H. Chan, C.-W. Ho, M. Nikolova, Salt-and-pepper noise removal by median-type noise detectors and detail-preserving regularization, Image Processing, IEEE Transactions on 14 (10) (2005) 1479–1485.

[18] P. Vincent, H. Larochelle, Y. Bengio, Extracting and composing robust features with denoising autoencoders, Proceedings of the 25th International conference on Machine Learning-ICML ’08 (2008) 1096–1103.

[19] V. Jain, S. Seung, Natural image denoising with convolutional networks, Neural information Processing Standard (2008) 1–8.

[20] F. Agostinelli, M. R. Anderson, H. Lee, Adaptive multi-column deep neural networks with application to robust image denoising, in: Advances in Neural Information Processing Systems, 2013, pp. 1493–1501.

[21] H. C. Burger, C. J. Schuler, S. Harmeling, Image denoising: Can plain neural networks compete with bm3d?, in: Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on, IEEE, 2012, pp. 2392–2399.

[22] K. Fotiadou, G. Tsagkatakis, P. Tsakalides, Low light image enhancement via sparse representations, in: Image Analysis and Recognition, Springer, 2014, pp. 84–93.

[23] X. Zhang, P. Shen, L. Luo, L. Zhang, J. Song, Enhancement and noise reduction of very low light level images, in: Pattern Recognition (ICPR), 2012 21st International Conference on, IEEE, 2012, pp. 2034–2037.

[24] Y. LeCun, F. J. Huang, L. Bottou, Learning methods for generic object recognition with invariance to pose and lighting, in: Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on, Vol. 2, IEEE, 2004, pp. II–97.

[25] J. Xie, L. Xu, E. Chen, Image denoising and inpainting with deep neural networks, in: Advances in Neural Information Processing Systems, 2012, pp. 341–349.

[26] A. santoso, E. Nugroho, B. Suparta, R. Hidayat, Compression ratio and peak signal to noise ratio in grayscale image compression using wavelet, International Journal of Computer Science and Technology 2 (2) (2011) 1–10.

[27] Z.Wang, A. Bovik, H.Sheik, E.Simoncelli, Image quality assessment: From error visibility to structural similarity, IEE Trans. Image Process. 13 (4) (2004) 600–612.

[28] P. Trahanias, A. Venetsanopoulos, Color image enhancement through 3-d histogram equalization, in: Pattern Recognition, 1992. Vol. III. Conference C: Image, Speech and Signal Analysis, Proceedings., 11th IAPR International Conference on, IEEE, 1992, pp. 545–548.

[29] H. Cheng, X. Shi, A simple and effective histogram equalization approach to image enhancement, Digital Signal Processing 14 (2) (2004) 158–170.

[30] F. Bastien, P. Lamblin, R. Pascanu, J. Bergstra, I. J. Goodfellow, A. Bergeron, N. Bouchard, Y. Bengio, Theano: new features and speed improvements, Deep Learning and Unsupervised Feature Learning NIPS 2012 Workshop (2012).

[31] J. Bergstra, O. Breuleux, F. Bastien, P. Lamblin, R. Pascanu, G. Desjardins, J. Turian, D. Warde-Farley, Y. Bengio, Theano: a CPU and GPU math expression compiler, in: Proceedings of the Python for Scientific Computing Conference (SciPy), 2010, oral Presentation.

[32] J. Bergstra, Y. Bengio, Random search for hyper-parameter optimization, The Journal of Machine Learning Research 13 (1) (2012) 281–305.