

A Novel Approach Using Convolutional Neural Network to Reconstruct Image resolution

Abdul Karim Armah¹, Michael Kwame Ansong¹, Samson Hansen Sackey¹, Ninjerdene Bulgan¹

¹College of IOT Engineering, Hohai University, 213022, Changzhou, China

DOI: 10.29322/IJSRP.9.05.2019.p8999

<http://dx.doi.org/10.29322/IJSRP.9.05.2019.p8999>

Abstract- Extensively, image super-resolution (SR) poses a challenge across all fields of interest as its problem is considered inherent in its acquisition due to several reasons. Hence, many algorithms have been proposed to suppress this inherent challenges. As a contribution to help see through this inherency, we modelled a Randomized Convolutional Neural Network for Image Super-Resolution (RCNNSR) which simply learns an end-to-end mapping existing between the low-resolution (LR) and the high-resolution (HR) and this reconstructed high-resolution image is kindred as possible with the corresponding ground truth high-resolution image. In image super-resolution, signal recovery is very necessary in order to get further information. Hence our tested images were totally exploit using the randomized leaky-rectified linear unit in order to compensate for information that lies in the negative region and therefore handling problem associated with over compression. To ensure fast convergence and also prevent large oscillation, the Nesterov's Accelerated gradient is used to improve convergence speed of the loss function. The algorithm is compared to other existing techniques and our proposed method averagely suppresses the other algorithms.

Keywords: Convolutional Neural Network Rectified Linear Unit, Deep Learning, Image Super-Resolution.

1. Introduction

Human security as well as the acquisition of detailed information of a person has made the image super-resolution (SR) reconstruction, especially in this 21st century a very active research field. Presently, this area of research strives to conquer the inherent limitations in devices that detect and convey information that constitutes an image or video. Smartphones and digital cameras are classical devices amongst the many that exhibit such inherency. Hence, this aids to make better use of the rapid growth of high-resolution (HR) display as it tends to ease accessing information in a form of image or video, ultimately for security reasons and other fantasies [1]. Image super-resolution could be thought of as a way of producing a high resolution (HR) image from one or multiple low resolution (LR) images through a well-structured (software technique) or written algorithm in any platform which can overcome the inherent limitations of low-cost imaging sensors. For its importance, such technology is applicable in medical imaging (x-ray radiography, etc.), surveillance (for monitoring human activities and events of a particular environment) and so on [2]. The basic reconstruction constraint of super-resolution is that the restored image, after applying the same generative model, should re-establish the observed low-resolution image. Mostly, this assumption is not realized simply because image super-resolution (SR) is usually an ill-posed inverse problem due to the insufficient number of low-resolution images, ill-conditioned registrations, unknown fuzzy operators, etc. Hence, these make the solution of the reconstruction constraint not unique. These ill-conditions create irregularities in the super-resolution (SR) image reconstruction. These setbacks are regularized into a well-posed problem by the integration of prior knowledge [1]. There are many approach to resolving these inverse problems but Bayesian super-resolution reconstruction method is usually utilized because of its robustness and flexibility in modeling noise [3].

Classification of Image Super-Resolution

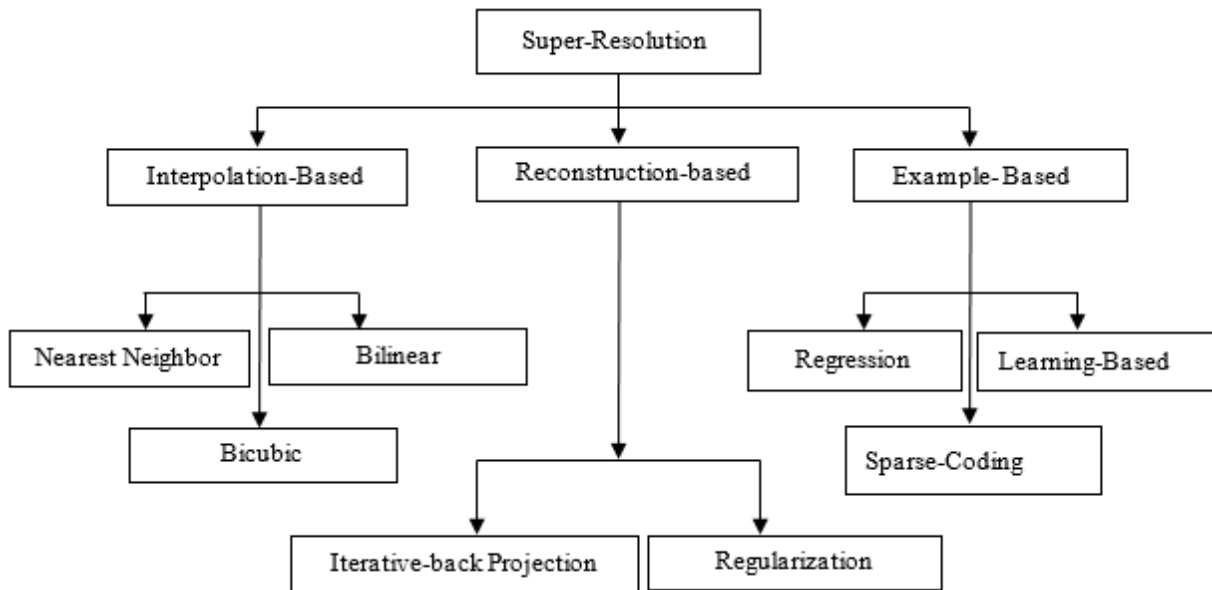


Figure 1: Flowchart of Super Resolution techniques.

Over the past years, researches concerning super-resolution (SR) have been made which had led to several proposed algorithms but the criteria upon which these algorithms are carried out can be grouped as shown above.

Interpolation Based Method: interpolation establishes a high-resolution image by calculating missing pixels' values and this approach can be implemented using traditional algorithms like nearest neighbor (NN), bilinear and bicubic interpolation. However, these interpolation-based methods commonly produce too smooth images as the magnification increase [4]. The most useful technique amongst the interpolation-based methods is the bicubic interpolation. It sharpens the edges and enhances the resolution of the image, but the disadvantage is that it produces ring- artifacts. To preserve and correct these artifacts, certain parameters (like local covariance coefficients) can be explored or algorithms (such as soft cuts algorithm) can be employed to check the regularization of the interpolation activities [5].

Reconstruction Based Method is divided into two (2) disciplines. These are; Iterative Back Projection Method (IBM) and the Regularization Method (RM).

Iterative Back Projection Method (IBM): It focuses on the blurriness of the image and the observed low resolution (LR) image in calculating the error difference of the HR image. This is usually done by back-projecting. For precision and accuracy, this procedure is repeated for a number of iterations to make the error insignificant [3].

Regularization Method (RM): The reason behind regularization algorithm is because SR reconstruction is an inverse and ill-posed problem. This is due to factors like unknown blurring operators, etc. Hence, various regularization methods such as Bayesian super-resolution reconstruction, total variation, sparsity and so on are the solutions to the ill-posed nature. This technique is to incorporate some prior knowledge about the desired HR image to constrain the solution space. The main advantage is that this method preserves edges and subdues artifacts in the resulting images [6].

Example-Based Method: The goal of this approach is to estimate the HR image using a dictionary corresponding patch. Here, the patch is constructed by either internal similarities or external training images. Since the micro-structural entities contained in the dictionary can be viewed as the image patches, it shows that the link between the HR image and LR image patches is established by the dictionary [5], [3]. Under the example-based method, one can adopt the technique of learning-based, sparse coding method or regression-based method.

Learning-Based Method: Basically, the estimation of the HR patches is from dictionary learning. Here, for every LR patch corresponds to a number of patches in the dictionary which is used in estimating the nearest neighbors. Afterward, the combination of the HR patches produces the HR image [3].

Sparse Coding Method: It employs two coupled dictionaries (D). Thus D_h for HR patches and D_l for LR patches. Coupling of these dictionaries enforce the sparse representation of the HR patches to equal the LR patches. The priority is training two coupled dictionaries for HR and LR image patches so that the sparse representation of the LR can be suitable for recovering the corresponding HR patches [5].

Regression-based method: the relationship existing between the LR and HR patches are studied by a regression function and the training sets are not learned from assumptions but rather the similarities within the LR and HR patches (thus, the correspondence of the LR and HR training sets). Now based on these training sets, the HR patches are obtained using the regression function by first minimizing the regularized cost function [3].

In this paper, we proposed a Randomized Convolutional Neural Network for Image Super-Resolution (RCNNSR) to reconstruct a high-resolution image from a low-resolution image and this reconstructed high-resolution image is kindred as possible with the corresponding ground truth high-resolution image.

The rest of the paper is arranged as follows: Section 2 of the paper is the literature review. Section 3 deals with the methodology which discusses convolutional neural network for image super-resolution and the techniques involved in each phase. It also explains the proposed method of the paper and the algorithm including the training. Section 4 elaborates on the experiment and the evaluations of the results by first describing the experimental parameters, the setups, our results and figure compared to other state-of-the-art. Section 5 of the paper is the last part which summarizes the study and concludes it.

2. Literature Review

Deep Learning

Deep learning is a derivative of machine learning of which its method is based on exploiting many layers of linear and nonlinear processes (in classification or in pattern analysis feature extraction and transformation) [7]. The architecture of deep learning comprises some inter-connected layers that are programmed in a way that the output of each layer is connected to the next layer's input. These transformation units make extraction of data (input or features) very simple. The convolutional neural networks (CNN) form the general deep learning architecture and the primary transformation unit of CNN during estimation is the convolution filters contained in the CNN's layers. The connectivity of these equally-sized layers permit the network to learn sets of features with different properties. Since the general sequence is the output been the input of the next layer and so forth, the hierarchy system in deep learning makes the learning of higher-level features at the last layers. Minimization of the error between the network output and the ground-truth output is basically learned through optimization models and these form the weight of the filters. One of the reasons making convolutional networks prominent in computer vision is its ability to exploit and also capture stationary characteristics and learn vital features in natural images [8].

As mentioned above, CNN is a deep learning technique or algorithm that has the ability to train many layer networks. The architecture of CNN is a multiplex structure but also very flexible with fewer parameters. Because the interconnection between the input, the hidden layers and the output layer that are incorporated in the network through shared weight, sparse coding, and other coding properties, it decreases the complexity regarding feature learning and training of data sets [9], [10]. Dong et al [11] proposed a deep learning technique from a deep convolutional network to estimate the high resolution of a single image. The mapping of the algorithm serves as a deep convolutional neural network which straightforwardly learns the end-to-end mapping between the low resolution and the high-resolution images. Also, expatiate how sparse coding-based method can be viewed as a deep convolutional neural network. The structure of the CNN used by Dong et al is lightweight which aid to speed up the estimation time. It can also simultaneously deal with three color channels to enhance both restoration and reconstruction quality. Cui et al [12] proposed an auto-encoder network and a model that upscale low-resolution images through deep network cascade (DNC). The DNC is not an end-to-end mapping between the low and the high-resolution images since each layer of the cascade requires independent optimization of the self-similarity search process and the auto-encoder [13]. The technique utilizes the non-local self-similarity in each layer to exploit the patches of the input image through the enhancement of the high frequency. The collaborative local auto-encoder (CLA) receive the enhanced image patches as input, which it then subdues the jaggy nature and enhance the overlapping capability of the patches before entering the next layer for the reconstruction of its super resolution.

Deep learning is a newly discovered method of machine learning and its popularity has made it the pivot or the spot amongst the research areas of neural network, artificial intelligence, graphical modeling, optimization, pattern recognition and signal processing due to the usage of large data sizes for training and current advances in machine learning and signals or information processing research. Since deep learning is applicable to different research fields, the methods of convolutional neural network have shown greater impact and perform very well in several applications related to computer vision by endowing the computer to visualize, identify, analyze, process and understand images in the same manner as humans and also produce suitable output [7].

3. Method

The reconstruction of a single image super-resolution in deep convolutional neural network follows a preprocessing measure before subjected to the training phase. The formulation of the process starts with interpolation. Let Y represents the low-resolution image and X be the ground truth high-resolution image. Here, the method of bicubic interpolation is used to resize the low-resolution image Y to attain the same size as X and the interpolated image Y now functions as the input of the fully-connected network. This allows the training stage to transform Y to the high-resolution image by learning the shared weight of different filters in the layers. Error minimization in the network related to the output and the original high-resolution image is also ensured. The main aim is to recover from the bicubic reconstructed image Y , an image $F(Y)$ closer to the ground truth high-resolution image X via a mapping function F which is characterized by both linear and non-linear relationships. The training phase comprises three systematic operations: the patch extraction and representation, the non-linear mapping and the reconstruction phase where the output is aggregated.

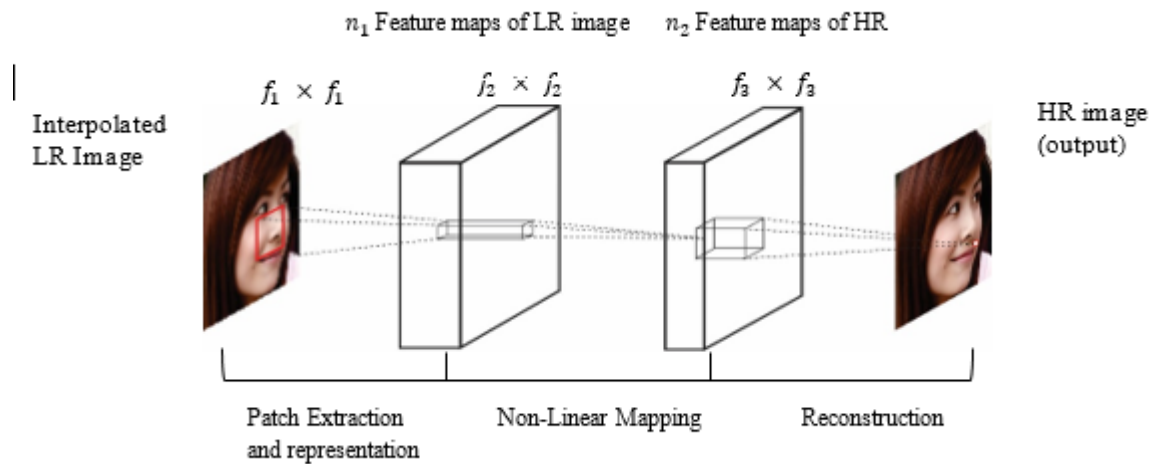


Figure 2: Illustration of a convolutional neural network

Feature Extraction and Representation Phase: This is the primary layer of the network which is also called the feature extractor layer. It is a phase that immediately accepts the interpolated image Y as input and operates on its features. In most image restoration techniques [14] patches are extracted and represented in a set of pre-trained manner such as Discrete Cosine Transform (DCT), Discrete Wavelet Transform (DWT) and Principal Component Analysis (PCA). These are Fourier related transforms that handle signals in their frequency domain and projecting images in their Eigen-space respectively. These pre-trainings form the basis of the feature extraction in the convolutional network where patches from the bicubic reconstructed image Y are extracted and each patch is represented or mapped into a high-dimensional vector. Here, the number of feature maps contained in the high-dimensional vectors equals the vectors dimension. The feature extraction and its representation into high-dimension space is purely linear convolution since we exclude the application of the rectified linear unit (ReLU) from the first layer operation and fully introduce it in the second layer to make the operation nonlinear. This implies the convolution of the image by a set of filters can be compared to the pre-trained bases such as DWT, DCT etc. in image restoration. The optimization of the network is based on the above mentioned bases. In the formulation, we defined the operation F_1 between the interpolated image Y and the first convolution layer, mathematically as:

$$F_1(Y) = \max(0, W_1 * Y + B_1), \dots \dots \dots (1)$$

Where W_1 denotes filters (weight). Also equals n_1 -filters of size $c \times f_1 \times f_1$. C represents the number of channels in the image Y . f_1 is the size of the spatial filter, n_1 denotes the filter number and B_1 stands for the biases of the filters and also the n_1 -dimensional vector where each element in the vector is connected to a filter. Therefore, the output of the extraction phase comprises the n_1 -feature maps.

Nonlinear Mapping Phase: The first layer operation deals with the extraction of n_1 -dimensional features for each patch which now serve as the input of the second layer for mapping. For successful convolutional computation, each of these n_1 -dimensional vectors extracted from the first layer is mapped into an n_2 -dimensional space. The output $F_2(Y)$ for the second layer is expressed as:

$$F_2(Y) = \max(0, W_2 * F_1(Y) + B_2), \dots \dots \dots (2)$$

Where W_2 is n_2 -filters of size $n_1 \times f_2 \times f_2$. B_2 represents the biases of size n_2 -dimension. The output of each n_2 -dimensional vector represents the high-resolution patches that will be utilized in the reconstruction phase. The function of this layer is to increasingly reduce the spatial size of the representation. This reduces the computational complexities because the number of parameters is decreased. Thereby, controlling over fitting.

Reconstruction Phase: Traditionally, image super-resolution reconstruction is achieved by aggregating the overlapping high-resolution patches to generate the final high-resolution image. This aggregation can be viewed as a pre-trained defined filter on a set of feature maps under the network. For this reason, the last convolutional layer for obtaining the final high-resolution image $F(Y)$ is given by:

$$F(Y) = W_3 * F_2(Y) + B_3 \dots \dots \dots (3)$$

W_3 represents c -filters of size $n_2 \times f_3 \times f_3$ which is also a linear filter. Also the size of B_3 is a c -dimensional vector. In the reconstruction phase, the filters play a specific role depending on the domain of the image. The filters behave as an averaging filter in the case where the high-resolution is in the image domain and can be reshaped and represented in a patch form. Otherwise in the case where the high-resolution patches are in other domain such as coefficients (in terms of other bases) the averaging is purely done by the W_3 by firstly projecting the coefficients into the image domain.

Rectified Linear Unit (RELU)

The rectified linear Unit (ReLU) function is an activation or ramp function which is proposed by Hahn Loesser et al. In Artificial Neural Network, the rectified linear function is also an activation function which defines and simplifies the positive part of an outcome. It suppresses the effectiveness of other rectifiers such as logistic sigmoid, the hyperbolic tangent, and so on due to its efficiency in biological and other application fields. Let the rectified linear function be a ramp function ($R(x): \mathbb{R} \rightarrow \mathbb{R}_0^+$). Analytically, it can be written in various forms. Let consider it as a system. This implies the system equation can be expressed as:

$$f(x_i) = \begin{cases} x_i, & \text{if } x_i > 0 \\ a_i x_i, & \text{if } x_i \leq 0 \end{cases} \dots \dots \dots (4)$$

And as a maximum function:

$$f(x) = \max(0, x) \dots \dots \dots (5)$$

Where x_i is the input image, a_i is the coefficient governing the slope of the negative part and i shows the variations in different channels that the nonlinear activation operates on. Based on the value or the characteristics of a_i , rectified linear function assumes either: ReLU (rectified linear unit), PReLU (Parametric- ReLU), LReLU (Leaky- ReLU) or RReLU (Randomized Leaky-ReLU) and these variations are the unique properties of the rectified linear function.

Now, if $a_i = 0$, it implies the ReL-function becomes ReLU and can be expressed as:

$$f(x_i) = \max(0, x_i) \dots \dots \dots (6)$$

If a_i = learnable parameter, then the function becomes PReLU, with the equation:

$$f(x_i) = \max(0, x_i) + a_i \min(0, x_i) \dots \dots \dots (7)$$

When $a_i = 0.01$ (small and fixed value), it is now an LReLU, with the support equation:

$$f(x_i) = \max(0, x_i) + a_i \min(0, x_i) \dots \dots \dots (8)$$

And finally, when a_{ij} is a random number sampled from a uniform distribution where $U(l, u)$ i.e $a_{ij} U \sim (l, u); l < u$ and $l, u \in [0; 1]$ [15].

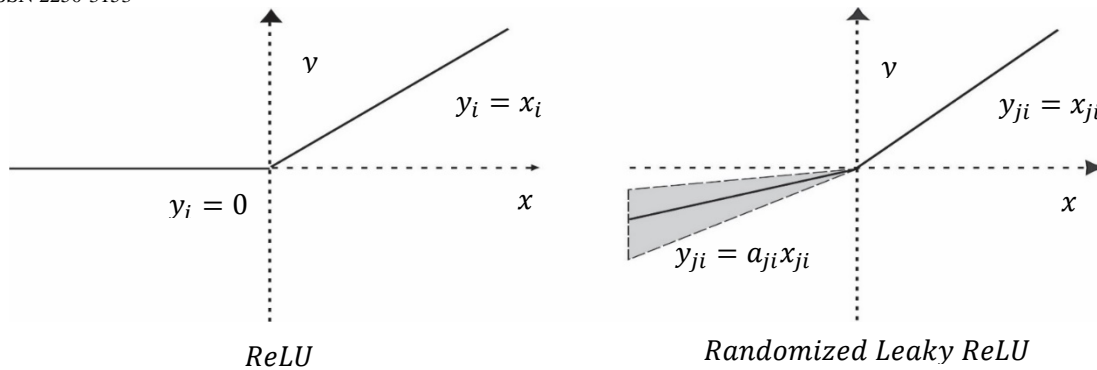


Figure 3: A characteristic graph between ReLU and RReLU

For RReLU, a_{ji} is a random variable which is sampled in a given range and remains fixed in testing. From equation (5), the maximum function applied to the convolutional neural network (CNN) takes the form $f(x) = \max(a_i, x_i)$, where $f(x_i)$ represents the final image associated with each layer's operation $F_i(x)$, $a_i = 0$ and $x_i = Wx + B$

Hence, the general activation function in CNN is expressed as:

$$F_i(x) = \max(0, Wx + B_i) \dots \dots \dots (9)$$

Equation (9) is a perfect example of ReLU since $a_i = 0$ and most of the algorithms related to CNN employ such activation function to structure the network but considering Fig.3, it shows that all the information is based or concentrated in only the quadrant $x > 0$ whilst other information in other quadrants are squashed to zero. Therefore, avoiding such a limitation and seeking to recover more information as possible, we employed the randomized leaky rectified linear unit (RReLU) as it extends to the lower quadrant $x < 0$ as in Fig.3 [16].

Hence from equation (5), the RReLU convolutional neural network system can be formulated as:

$$y = \begin{cases} W_1x + b_1, & \text{if } x > 0 \\ \frac{W_2x + b_2}{a}, & \text{if } x < 0 \end{cases} \dots \dots \dots (10)$$

Where a assumes the same conditions as a uniform distribution. Combining equation (9) and (10), the final RReLU formulae can be generalized as:

$$y = \max(0, W_1x + b_1) + \frac{\min(0, W_2x + b_2)}{a} \dots \dots \dots (11)$$

Now equation (11) is integrated into the weights (filter response) of the various layers of the network with the same parametric properties during the formulation process. The new equations of the extraction and representation phase, the nonlinear mapping phase and the reconstruction phase are as follow respectively:

$$F_1(Y) = \max(0, W_1 * Y + B_1) + \frac{\min(0, W_1 * Y + b_1)}{a} \dots \dots \dots (12)$$

$$F_2(Y) = \max(0, W_2 * F_1(Y) + B_2) + \frac{\min(0, W_2 * F_1(Y) + B_2)}{a} \dots \dots \dots (13)$$

$$F(Y) = W_3 * F_2(Y) + B_3 \dots \dots \dots \text{(same as equation 3)}$$

ReLU normally paralyzes or deactivates the neuron by putting it to zero when a negative value arises but the RReLU allows some small negative values. However, for consistency and comparison we maintained the procedure proposed by Srivastava et al [17] in training, as in the method of dropout by aggregating all a_{ij} and setting it to $\frac{l+u}{2}$ in the testing phase to obtain a deterministic result.

Also maintained the sampling parameters constant as suggested by Kaggle National Data Science Bowl (NDSB) competition that due to the randomized nature of the RReLU, it could reduce over fitting by sampling a_{ij} from $U(3, 8)$. This implies $a = \frac{11}{2}$.

Training

The training phase requires an end-to-end mapping function F for estimating the network parameters. Let the network parameter $\theta = \{W_1, W_2, W_3, B_1, B_2, B_3\}$ where W_i and B_i are the weights and the biases of the various layers respectively. For better performance, the Mean Squared Error (MSE) is used as the loss function since it favors a high peak-to-signal- noise ratio (PSNR).

Let $\{X_i\}$ be a set of high-resolution images and $\{Y_i\}$ be its associated low-resolution images. hence the loss function can be expressed as:

$$L(\theta) = \frac{1}{n} \sum_{i=1}^n \|F(Y_i; \theta) - X_i\|^2 \dots \dots \dots (14)$$

where $F(Y; \theta)$ is the reconstructed image, X represents the ground truth high-resolution images and n is the number of training samples. The MSE is an important function mostly used for quantitatively evaluating image restoration qualities as its value is very significant in assessing the quality of image resolutions. There are many loss functions for training both traditional methods and CNN but if the loss function is flexible and derivable, the network can adapt to its parameters as compared to the challenges other methods would exhibit [13]. Although the framework is in the direction of favoring the PSNR, it also favors other image restoration metrics such as the structural similarity index measure (SSIM), Weighted Peak-To-Signal Noise Ratio (WPSNR), Mean Structural Similarity Index Measure (MSSIM) and so on. The minimization of the loss function by the various gradient descents by updating the weight W of the network is one of the integral part of evaluating how good or adaptive an algorithm is, and its simulation power (executing faster within a shorter time). In most cases or classically, training a neural network is minimizing the loss function by backpropagation, which is a specific gradient descent method for the neural network but recently, stochastic gradient descent (SGD) which is used for approximating the precise gradient when dealing with a large dataset is developed [18]. The SGD is an optimization technique or algorithm for modeling the parameters of the network in an iterative manner. The first iteration is to adjust the parameters a bit, and successively is to keep adjusting the parameters until it reaches a point that will enhance the performance of the network [19].

Consider an update on the weight matrices using the iterative equation:

$$\Delta_{i+1} = \mu \Delta_i + \eta \nabla L(W_i^l) \dots \dots \dots (15)$$

$\mu \in [0,1]$, if $\mu = 0.9$ and $\nabla L(W_i^l) = \frac{\partial L}{\partial W_i^l}$, it implies

$$\Delta_{i+1} = 0.9 \cdot \Delta_i + \eta \cdot \frac{\partial L}{\partial W_i^l} \dots \dots \dots (16)$$

$$W_{i+1}^l = W_i^l + \Delta_{i+1} \dots \dots \dots (17)$$

The SGD algorithm updates the weight W_{i+1} of the network through classical momentum (CM) by the linear combination of the previous update Δ_i and the derivative $\nabla L(W_i^l)$ or $\frac{\partial L}{\partial W_i^l}$. The momentum $\mu \in [0,1]$ is the weight of the previous update and mostly chosen to be 0.9, Δ_{i+1} represents the value for estimating the weight at $i+1$, l is the layers and since it follows the SRCNN architecture, it implies $l \in \{1,2,3\}$ and i is the iterations. Although SGD converges faster, it does not involve storing the overall training data in memory. It is flexible when adding new data in an online setting. SGD is used for analyzing loss functions in their convex form [20]. In deep learning, this loss function is non-convex to the parameters of the network, therefore, there is no assurance that it can estimate the global minimum, although it gives a good solution to deep learning networks. Hence we used the techniques of Nesterov’s Accelerated Gradient (NAG) [44] to update the weights of the network.

Let the NAG formulae be:

$$\Delta_{i+1} = \mu \Delta_i + \eta \nabla L(W_i^l + \mu \Delta_i) \dots \dots \dots (18)$$

$$W_{i+1}^l = W_i^l + \Delta_{i+1} \dots \dots \dots (19)$$

Comparing SGD and NAG, the difference is in the derivative part. When updating, the SGD updates the weights through the linear combination of the previous update and the derivative whilst NAG updates the weights by firstly executing a partial update on W_i^l . Estimate $W_i^l + \mu\Delta_i$ which when considering equation (19) is W_{i+1}^l . This enables the Nesterov's Accelerated Gradient to respond better in regenerating Δ in a faster manner and in the situation where μ is higher, it becomes more stable than classical momentum. In the case where the optimization path of the classical momentum shows large oscillations in the direction vertical to its high-curvature, NAG has the ability to prevent these oscillations completely. This property makes it much efficient in decelerating during multiple iterations and also very tolerant to large values of μ compared to CM [17]. Now on each layer, the weights of the filters are determined by randomly drawing from a Gaussian distribution with zero-mean, a standard deviation of 0.001 and zero (0) biases. The learning rate of the first two layers and that of the last layer are 10^{-4} and 10^{-5} respectively. In the training phase, instead of patches, sub-images (samples considered as small images) were used and therefore it does not require any later processing such as averaging. Hence, the preparation of the ground truth images $\{X_1\}$ is set as $f_{sub} \times f_{sub} \times C$ pixel sub-images that are cropped randomly from the trained images. For the LR samples $\{Y_i\}$, the sub-images were blurred using Gaussian Kernel and sub-sampling the blurred image by an up-scaling factor using bicubic interpolation. Now to prevent border effects in the training phase, padding is excluded in all the convolutional layers enabling the network to create smaller output $[(f_{sub} - f_1 - f_2 - f_3 + 3)^3 \times C]$, and the evaluation of the MSE loss function is solely based on the difference between the central pixels X_i and the network output.

Algorithm

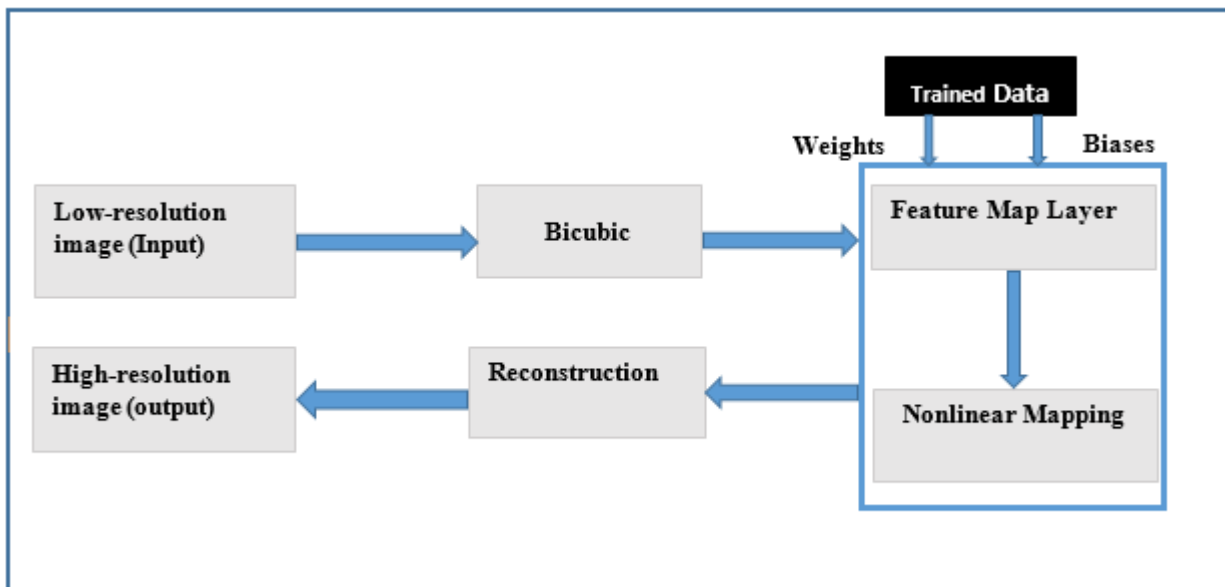


Figure 4: Proposed Method Workflow

Ground truth high-resolution image X

Input: Low-resolution (LR) image Y

Interpolate the LR image Y to the same size as X

The feature extraction and representation phase (convolution):

$$F_1(Y) = \max(0, W_1 * Y + B_1) + \frac{\min(0, W_1 * Y + b_1)}{a}$$

The nonlinear mapping phases

$$F_2(Y) = \max(0, W_2 * F_1(Y) + B_2) + \frac{\min(0, W_2 * F_1(Y) + B_2)}{a}$$

Image reconstruction phase (convolution):

$$F(Y) = W_3 * F_2(Y) + B_3$$

Output: High-resolution (HR) image same as X .

4. Experiment

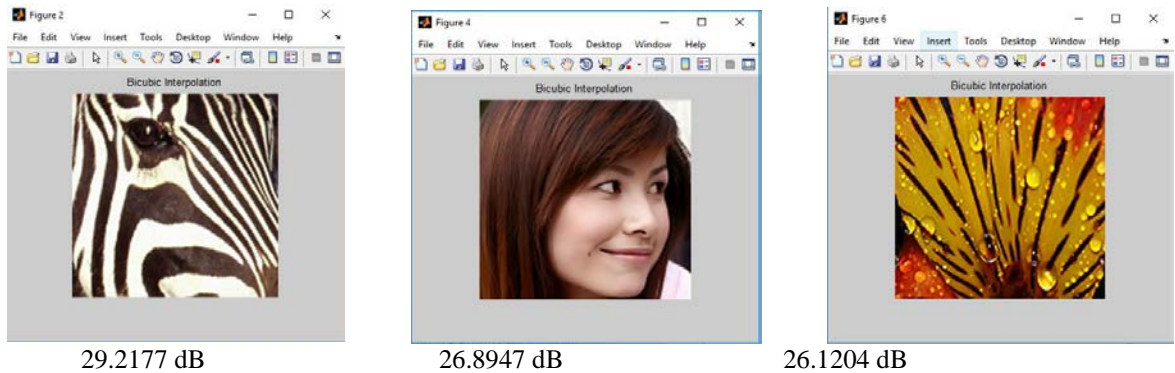
<http://dx.doi.org/10.29322/IJSRP.9.05.2019.p8999>

www.ijsrp.org

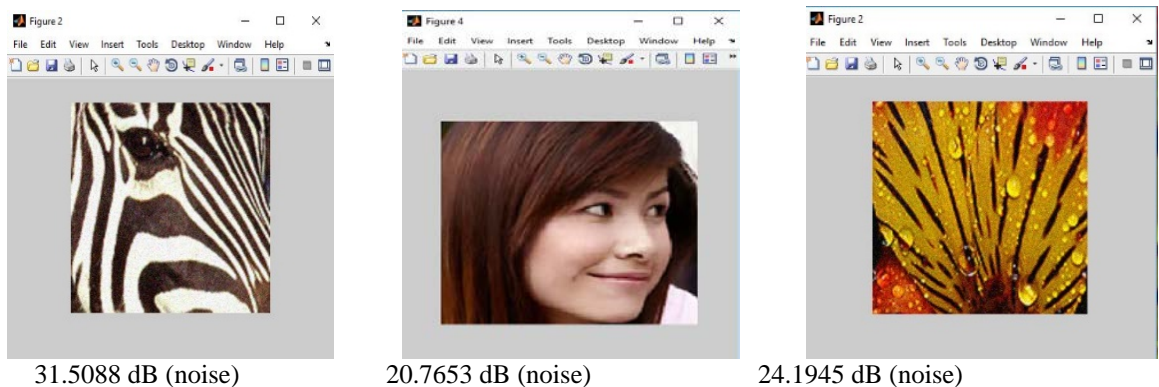
This study is motivated by the image super-resolution using convolutional neural network (SRCNN) algorithm. Hence the experimental setups follow a three-layer SRCNN architecture and parametrically as adopted by Dong et al [11]. The supremacy of the algorithm is tested by comparing it to other state-of-the-art such as sparse coding (SC) proposed by Yang et al [1], bicubic method, image denoising via principal component analysis with local pixels grouping (LPG-PCA) by Zhang et al [21] and other known methods of which their techniques are relevant and the codes to their models are publicly made available by the authors. Our novel named Randomized Convolutional Neural Network for Image Super-Resolution (RCNNSR) is evaluated using Caffe package configured through visual studio 2015. For feature extraction this is linked to a MATLAB platform to extract the features. Selections of fewer images (such as the zebra, lady, flower, etc.) from the dataset were used for evaluations and other computations such as the peak signal to noise ratio (PSNR). Following the setup of SRCNN, we set the filters as $f_1 = 9, f_2 = 5, f_3 = 5$, thus, $(9 - 5 - 5)$ and $n_1 = 64, n_2 = 32$. Also set $\mu = 0.9$ [22] and the up scaling factor for bicubic interpolation is considered as 3 for the training of the network. The initialization of the filter weights for each layer is a zero (0) mean and a standard deviation of 0.001 randomly derived from a Gaussian distribution, with zero (0) biases. The learning rates set for the layers of the network are 10^{-4} for the first two layers and 10^{-5} for the third layer. The training phase establishes the preparation of the ground truth images $\{X_i\}$ of pixel-dimension 32×32 pixel sub-images. The sub-images are cropped in a random manner from the training images. Now, these sub-images are samples considered as small images instead of patches. Such a technique is employed to prevent any post-processing such as averaging of overlapping patches. Also, the LR sample $\{Y_i\}$ is prepared by blurring the sub-image by Gaussian Kernel. This is then sub-imaged using bicubic with the same up scaling factor. Comparatively, we employed the training set, protocols, and the test set as described in [22]. 91 images were considered as the training set. Within the training set, we selected three (3) images for evaluation purposes with the upscale factor 3. We set $c = 1$ in the first layer and that of the last layer for the luminance channel (in YCbCr color) which is in accordance with other relevant traditional methods relating to luminance channels. Ensuring the same input and output image sizes, and eliminating border effect in the training phase, the layers of the network were structured with adequate zero-padding during the testing phase.

Simulation Results

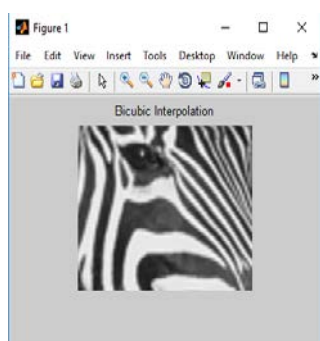
Bicubic output for Sparse Coding (SC)



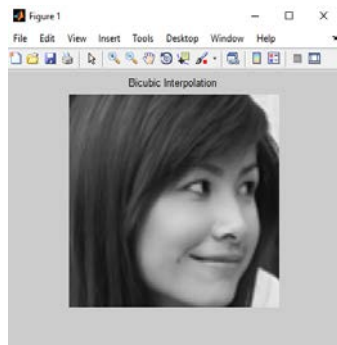
Bicubic output for Principal Component Analysis with Local Pixels Grouping (LPG-PCA)



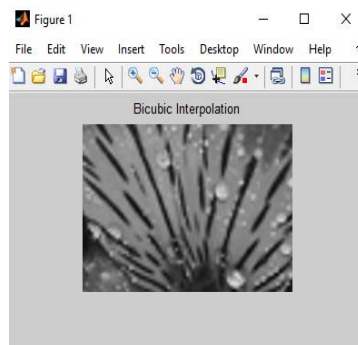
Bicubic output for Randomized Convolutional Neural Network for Image Super-Resolution (RCNNSR)



22.643732 dB

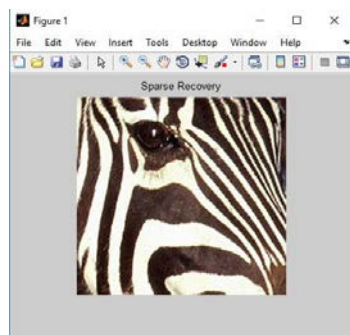


32.562617 dB

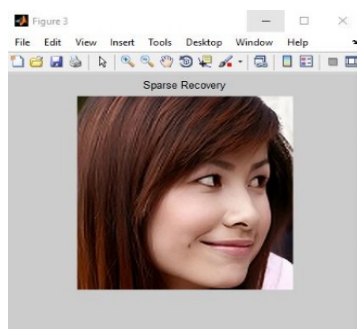


24.323178 dB

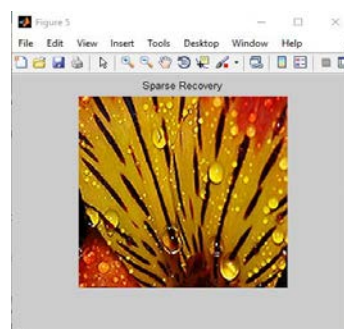
Simulation Results for Sparse Coding (SC)



29.3898 dB

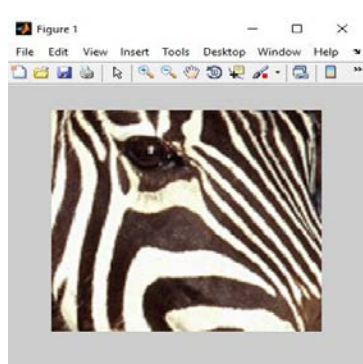


26.9301 dB

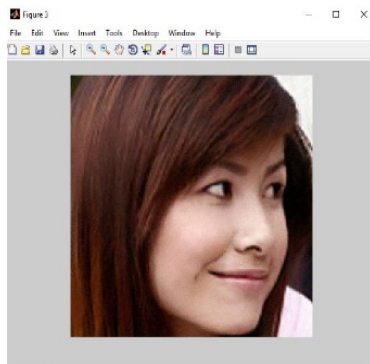


26.3030 dB

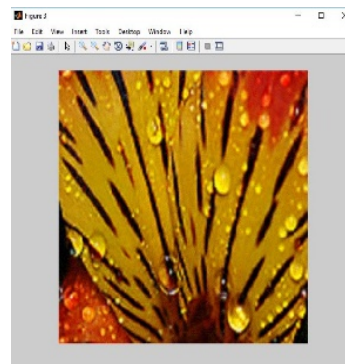
Simulation Results for Principal Component Analysis with Local Pixels Grouping (LPG-PCA)



26.4232 dB

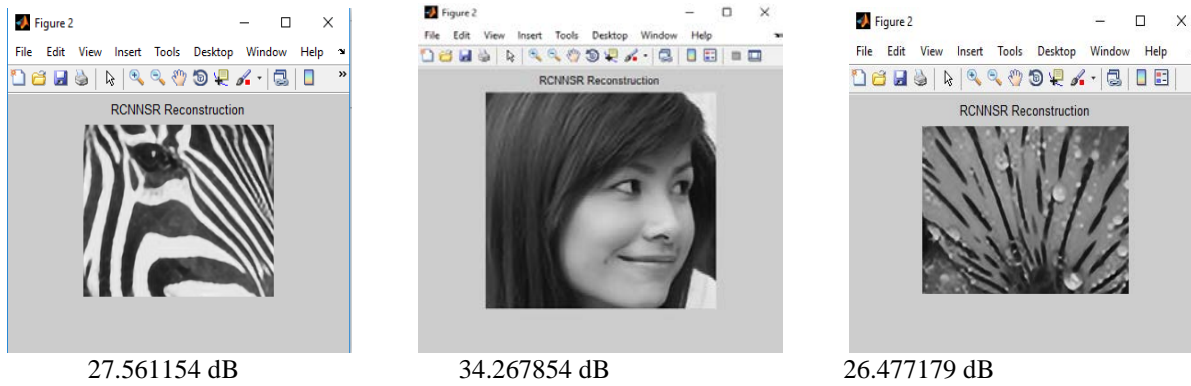


33.6703 dB



25.4995 dB

Simulation Results for Randomized Convolutional Neural Network for Image Super Resolution (RCNNSR)



Evaluations

Table 1: The result of 3-images in PSNR (dB) selected from 91 trained datasets

Image \ Technique	RCNNSR	RCNNSR Cubic	SC	SC Cubic	LPG-PCA	LPG-PCA (Noise)
Zebra	27.5612	22.6437	29.3898	29.2177	26.4232	20.7653
Lady	34.2679	32.5626	26.9301	26.8947	33.6703	32.5988
Flower	26.4772	24.3232	26.3030	26.1204	25.4955	24.1945
average	29.4354	26.5100	27.5410	27.4109	28.5297	26.1371

Table 1 show the results of our algorithm (RCNNSR) compared to other state-of-the-art. We calculated the PNSR and the bicubic value for each image used for the experiment is 3. With respect to the other two algorithms (Sparse coding and LPG-PCA) we compared each peak signal to noise ratio (PSNR) values of the selected images. Now based on the result, our algorithm produces a satisfactory outcome and averagely outperforming the other methods. Comparatively, the LPG-PCA (PSNR value) is much closer to ours model (RCNNSR) with a difference of 1.138dB (Zebra), 0.5976dB (lady) and 1.2391dB (flower). Also, in each simulation, we considered the bicubic outputs of the various images and their corresponding estimates. The bicubic algorithm is one of the common techniques in image super-resolution and although it is better than other interpolation-based methods, it produces jaggy artifacts. Even with such defect in the images which can be analyzed by inspection, the RCNNSR bicubic images are much sharper and of good quality compared to the sparse coding and the LPG-PCA (noise) although the SC cubic is higher for the zebra and the flower. With the sparse coding, our algorithm outperformed the sparse output by 7.3378dB (lady), 0.1742dB (flower) but the sparse coding PSNR value for the zebra outperformed our model by 1.8286. Although the simulation time is not a priority, the LPG-PCA algorithm takes much time during execution and followed by SC. Generally, the PSNR value shows how quality the image is, and vice versa. Hence, on the average, our algorithm RCNNSR shows a higher PSNR value (29.4354dB) followed by the LPG-PCA (28.5297dB) and SC (27.5410dB).

5. Conclusion

Conclusively, we have provided a deep learning technique for image super-resolution. Our proposed Randomized Convolutional Neural Network for Image Super-Resolution (RCNNSR) method cements an end-to-end mapping learning by totally exploiting the nonlinearity existing between the low-resolution images and the high-resolution images. We seek to recover more information of the images as possible, hence we utilized the randomized leaky rectified linear unit as an activation function to handle the problem of over compression because the rectified linear unit simply set or squash any value (information) below the $x > 0$ quadrants to zero. In updating the weights of the deep convolution neural network, the optimization path of the classical momentum shows large oscillation which is prevented by employing the Nesterov's Accelerated Gradient and also to aid facilitate the loss function convergence. Comparatively, sparse representation methods operate by handling separately each component whilst our technique jointly optimizes all the layers in the network. However, the supremacy of our algorithm is tested by comparing it to other state-of-the-art to know how to further improve the technique.

Reference

- [1] J. Yang, J. Wright, T. Huang, and Y. Ma, "Image Super-Resolution via Sparse Representation", IEEE Transactions on Image Processing, Vol. 19, No. 11, pp. 1, November 2010.
- [2] X. Li, H. He, R. Wang, and D. Tao, "Single image super-resolution via directional group sparsity and directional features", IEEE Transactions on Image Processing, Vol. 24, No. 9, pp. 1, September 2015.
- [3] J. Dalvadi, "A Survey on Techniques of Image Super Resolution", International Journal of Innovative Research in Computer and Communication Engineering, Vol. 4, Issue 3, pp.2-5, March 2016.
- [4] Q. Zhu, L. Sun and C. Cai, "Image Super-Resolution Via Sparse Embedding", IEEE World Congress on Intelligent Control and Automation (WCICA), pp. 1-4, 2014.
- [5] X. Lu, Y. Yuan and P. Yan, "Alternatively Constrained Dictionary Learning for Image Super-resolution", IEEE Transactions On Cybernetics, Vol. 44, No. 3, pp. 1-3, March 2014.
- [6] F. Cao, M. Cai, Y. Tan, and J. Zhao, "Image Super-Resolution via Adaptive $\ell_p(0 < p < 1)$ Regularization and Sparse Representation", IEEE Transactions On Neural Networks And Learning Systems, Vol. 27, No. 7, pp. 1-2, July 2016.
- [7] L. Deng and D. Yu, "Deep Learning Methods and Applications", Foundations and Trends in Signal Processing Vol. 7, pp. 1-10, 2014.
- [8] A. Ducournau, and R. Fablet, "Deep Learning for Ocean Remote Sensing: An Application of Convolutional Neural Networks for Super-Resolution on Satellite-Derived SST Data", Conference: 9th Workshop On Pattern Recognition in Remote Sensing pp.1-6, 2016.
- [9] M. H. Lim, and Y. S. Ong, "Proceeding in Adaption Learning and Optimization," ISSN: 2363-6084, 2015.
- [10] Y. Tang, X. Zhu and M. Cui, "Convolutional Neural Network with Gradient Information for Image Super-Resolution", Proceedings of the IEEE International Conference on Information and Automation, pp. 1-6 August 2016
- [11] C. Dong, C. C. Loy, K. He, and X. Tang. "Learning a deep convolutional network for image super-resolution". In Proceedings of the European Conference on Computer Vision (ECCV), pp. 1-5, 2014.
- [12] Z. Cui, H. Chang, S. Shan, B. Zhong, and X. Chen, "Deep network cascade for image super-resolution," in Proc. Eur. Conf. Comput. Vis., pp. 49–64, 2014.
- [13] C. Dong, C. Loy, K. He and X. Tang, "Image Super-Resolution Using Deep Convolutional Networks", IEEE Transactions On Pattern Analysis and Machine Intelligence, Vol. 38, No. 2, Pp. 1-13, February 2016.
- [14] M. Aharon, M. Elad, and A. Bruckstein, "K-SVD: An algorithm for designing overcomplete dictionaries for sparse representation," IEEE Trans. Signal Process., vol. 54, no. 11, pp. 1-7, Nov. 2006.
- [15] Jefkine. "Formulating the ReLU". Internet: <http://www.jefkine.com/general/2016/08/24/formulating-the-relu/>, August 24, 2016.
- [16] X. Yang, S. Zhang, C. S Hu, Z. C. Liang, and D. D. Xie, "Dropout: Super-resolution of medical image using representation learning", 8th International Conference on Wireless Communications & Signal Processing (WCSP), Pp.1-6, 2016.
- [17] N. Srivastava, G. E. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "A simple way to prevent neural networks from overfitting". The Journal of Machine Learning Research, 15(1):1929-1958, 2014.
- [18] R. Vidal, J. Bruna, R. Giryes, and S. Soatto, "Mathematics of Deep Learning", Cornell University, Machine Learning, 2017.
- [19] http://wiki.fast.ai/index.php/Gradient_Descent#Stochastic_Gradient_Descent
- [20] M. Schmidt, N. Le Roux, and F. Bach. "Minimizing finite sums with the stochastic average gradient". *Mathematical Programming B*, Springer, 162 (1-2), pages 83–112, 2017.
- [21] L. Zhang, W. Dong, D. Zhang, and G. Shi, "Two-stage image denoising by principal component analysis with local pixel grouping". *Pattern Recognition*, 43(4), pp. 1531-1549. 2010.
- [22] X. L. Li, Y. T. Hu, X. B. Gao, D. C. Tao, and B. J. Ning, "A Multi frame image super restoration method", *Advances in Signal Processing*, Vol. 90, Issue 2, February, pp. 405-414, 2010.

AUTHORS

Abdul Karim Armah, Graduate Student, Hohai University, College of IoT Engineering, 213022, China, armahabdulkarim1@yahoo.com

Michael Kwame Ansong, Graduate Student, Hohai University, College of IoT Engineering, 213022, China, ansongkwame@gmail.com

Samson Hansen Sackey, Graduate Student, Hohai University, College of IoT Engineering, 213022, China, samsonsackey@yeah.net

Ninjerdene Bulgan, Graduate Student, Hohai University, College of IoT Engineering, 213022, China, b_ninje@yahoo.com