

# Deeping Learning the Transformation Process for Fast Image Enhancement

Mengcheng Xiang\*, Jinguang Cheng\*\*

\*(The Second Senior High School, Sanming City, Fujian Province, China

Email: 89983777@qq.com)

\*\* (Department of Electronic and Information, College of Qiuzhen, University of Huzhou University, Huzhou, China

Email: 136966885@qq.com)

\*\*\*\*\*

## Abstract:

We propose a novel deep learning-based image enhancement method which learn the transformation function or matrix instead the image content to achieve fast enhancement of the image. The structure of the Network used in this work utilizes four residue blocks of the ResNet-18 as the basenet to train the transformation function or matrix of the input image  $F$ , and outputs the transformation matrix  $H$ , thus this net is also named FHNet. With the learned information contained in the transformation matrix  $H$ , the enhancement of the image feature is easily implemented with multiplying the transformation matrix with the input images, that is  $G=HF$ . This method is proved to be effective for the feature enhancement of the image, and which can process the image enhancement with fast speed and high quality. Compared with conventional method, our proposed method can effectively solve the problem of color distortion and characters with small memory consumption, low computation cost and high speed.

*Keywords* — **Image Enhancement, Deep learning, FHNet.**

\*\*\*\*\*

## I. INTRODUCTION

Image enhancement is an important branch of digital image processing, which are important in many fields, such as precision medical [1-3], nanoscale light element identification [4], Robot [5], Fault and Fraction Network identification [6], magnetic resonance imaging [7-9], 3D printing [10], Earth Observing [11], Underwater Animal Detection and Classification [12], Optical Microscopy [13], image processing [14-17]. Due to the influence of scene conditions, the visual effect of image shooting is not always good, which requires image enhancement technology to improve human visual effect, such as highlighting some characteristics of the object in the image, extracting the characteristic

parameters of the object from the digital image, etc., which are conducive to the recognition, tracking and understanding of the target in the image. The main content of image enhancement is to highlight the interested part of the image and weaken or remove unnecessary information. In this way, the useful information can be enhanced, so that a more practical image can be obtained or converted into an image more suitable for human or machine analysis and processing.

In fact, there have been lots of enhancement algorithms reported for the image contrast enhancement, image noise reduction and image brightness improvement for low-light images [18]. Among the algorithms, histogram equalization (HE) algorithm is a typical approach [19], which simply reassigns the distribution of the different value of

image pixels to make them uniform distribution. However, HE algorithm usually loss image details. To deal with this problem, another enhancement algorithm based on the Retinex theory are proposed. The Retinex-based algorithms separate the image into illumination part and reflection part and process them with Gaussian low-pass filtering and logarithmic transformation respectively [20]. However, these algorithms always lead to unpleasant artifacts to the final results. In addition, enhancement algorithms with dehazing process [21], can give a good result but usually lead to the images over enhanced and exaggerated.

In recent decade years, there have developed many methods based on deep learning or machine learning framework [22–30]. These approaches are proved to be effective and superior to previous traditional algorithms in multiple areas[31-34].

We present a new network architecture, named FHNet, that is learning the image transformation process instead the image itself, to achieve the fast enhancement of the input images with fast speed and high quality.

## II. FHNET ARCHITECTURE

### A. Conventional image enhancement method

Conventional image enhancement methods can be divided into two categories, e.g. spatial domain method and frequential domain method. The former imposes two-dimensional Fourier transformation on the images to filter the noise with low-pass filtering or enhance edges or profiles with high pass filtering to make the blurred objects clear. The latter usually executes spatial filtering that performs local average operation with specially designed filtering template to remove or weaken noise or to outstand the object features.

In spatial domain, the image enhancement method can be mathematically described as

$$g(x, y) = f(x, y) * h(x, y) \quad (1)$$

where  $f(x,y)$  is the original image,  $g(x,y)$  is the output image,  $h(x,y)$  is the transformation function of the spatial domain with  $x$  and  $y$  be the spatial coordinates or index number of the individual pixel of the image. The symbol “ $*$ ” take the meaning of convolution that are a two-fold integration across the image place.

Similarly, in frequential domain, image enhancement operation is implemented by multiplying the Fourier transformation of the original image with a desired frequential filter, that is

$$G(u, v) = F(u, v) * H(u, v) \quad (2)$$

where  $F(u, v)$  is the Fourier transformation of  $f(x,y)$ ,  $H(u, v)$  is the frequential filter, and  $G(u, v)$  is the frequential domain counterpart of  $g(x,y)$ .

The key point of the convention image enhancement method is the design of the spatial transformation function or its Fourier transformation. Usually, these functions, such as gamma transform, histogram equalization, contrast or brightness stretching, etc, are difficult to determine due to their parameters are case dependent necessitating individual adjustment according to relevant experience, which leads to poor performance to the generalization and robustness of them, and be easy to cause over fitting phenomenon.

### B. The idea of FHNet based image enhancement

To overcome the above-mentioned shortcomings in the conventional image enhancement method, this paper proposes a novel image enhancement method based on the deep learning method, which inherit the idea of the conventional method but equip with the deep learning method.

Similar to the conventional image enhancement method, this proposed image enhancement method processes the original image with some transformation (linear transformation or nonlinear transformation), which can be mathematically described as:

$$G = H \cdot F \quad (3)$$

here  $F$  represents the original input,  $H$  is the transformation matrix. Different from the classical method where the input is the original image or its Fourier transformation, the input  $F$  in above equation is not the original image or its Fourier transformation, but its convolution with four-layer deep learning neural Net.

This paper constructs a full convolution Network with the deep learning method to learn the transform matrix  $H$  at low lever resolution picture. For the convenience, in this paper, the proposed convolution neural Net is thus called as FHNet.

### III. MODEL CONSTRUCTION

#### A. The basenet of FHNet architecture

The structure of our proposed FYNet is shown in Fig.1. It can be seen that from Fig.1, FYNet can be considered as the combination of two sub Nets, one is FNet that utilize ResNet architecture related with the input image F, the other is the HNet that related with the transform matrix H.

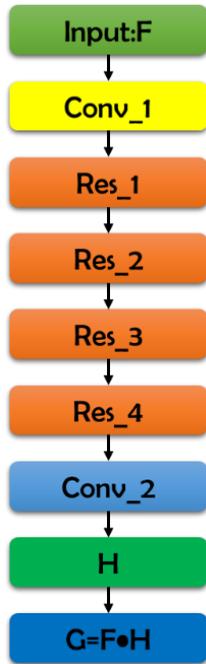


Fig.1 The architecture of FHNet

The FNet utilize basic unit of the ResNet to extract the feature information of input image since it is easy to modify and extend without the worry of Net performance degradation given that the training data is enough and the network is deepened gradually.

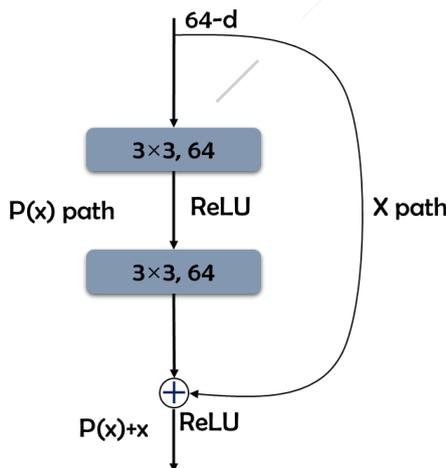


Fig.2. The basenet of the FHNet

The architecture of the basic unit of the ResNet is called residue blocks, of which the structure can be seen in the Fig.2. In this paper, there are 4 residue blocks. The structure of residue block in this paper can be seen in Fig.2. Each residue block has two paths: one is  $G(x)$  path, the other is  $x$  path that is typically called as shortcut path. The addition symbol in the Fig.2 is the element-wise addition, requiring that the size of the element that involved in the calculation is the same. The core problems related with the residue block lies in three aspects:(1)the design of residue path;(2)the design of the shortcut; and (3)the connection of the residue blocks.

For the design of residue path. Each residue block consisting of at least two or more convolution layers, is fit to a fitting function  $G(x)$ , with  $G(x):=P(x)-x$ , where  $P(x)$  is an expected potential mapping,  $G(x)$ , and  $x$  be the sample. Obviously,  $G(x)$  is the residue of the expected potential mapping  $P(x)$  and  $x$ . Therefore, instead of directly searching the expected potential mapping  $P(x)$ , residue blocks can learn to push the residue  $G(x)$  to zero, rather than fit an identity mapping  $P(x)=x$  by a stack of linear or nonlinear layers. This path is also called  $P(x)$  path as seen in Fig.2. Actually, according to difference in the structure, the residue path can also be classified into two categories: bottleneck block and basic block. The bottleneck block includes two convolution layers, which is used to firstly reduce and then increase the calculation dimension to reduce the computation complexity; and the basic block does not include bottleneck, instead it only contains two convolution layers. In this paper, we select the basic block as the residue path due to its simple structure as seen in Fig.2.

Similarly, shortcut can also be classified into two classes in the light of the size and the quantity of the feature map are changed or not by the residue path. One kind of shortcut keeps the  $x$  unchanged when it is output to the element-wise adder; the other kind is instead necessary to modify the dimension of the sample to match the size or shape of the residue path. In this paper, to make information can be smoothly transmitted in forward and backward propagation, we keep the "purity" of the shortcut path by moving the rectified linear unit (ReLU) to the residue path

and as far as possible decreasing the use of convolution operation.

As the connection of the residual blocks, after a rectified linear unit, the  $G(x)+x$  is directly used as the input  $x$  of the next residue block. In addition, the shortcut of the next residue block is simply connected with the input  $x$  with identity.

This proposed ResNet-18, compared with plain net[35,36] and VGG-19 net[37], which has no shortcuts in their networks, ResNet-18 which has 18 residue blocks, has more control dimension to avoid the network performance “degradation”, for example, the shortcut paths alter the back propagation from multiplication to addition so that the final loss of the network in the back propagation can be lossless direct to each block, meaning that the weight update of each block directly affects the final loss. Compared with the basic units of the VGG-19 net[37], the residue blocks needs not a pooling layer for each residue block for the data scale reduction, instead the final desired feature can be sufficiently obtained with a global average pooling layer rather than with a full connection layer, thus greatly improves the processing efficiency.

Detailedly, let  $x_l$  and  $x_L$  be the respective input of the  $l$ th block and  $L$  the block, the relationship of them can be mathematically obtained as[35,36]:

$$x_L = x_l + \sum_{i=1}^L F(x_i, w_i) \quad (4)$$

where  $x_i$  and  $w_i$  are the input and weight of the  $i$ th layer. The back propagation relationship can be figured out as[36]:

$$\frac{\partial \varepsilon}{\partial x_l} = \frac{\partial \varepsilon}{\partial x_L} \frac{\partial x_L}{\partial x_l} = \frac{\partial \varepsilon}{\partial x_L} \left( 1 + \frac{\partial}{\partial x_{i-1}} \sum_{i=1}^L F(x_i, w_i) \right) \quad (5)$$

with  $\frac{\partial \varepsilon}{\partial x_l}$  be the back-propagation gradient. The “1” in the back propagation has a good property, which can effectively avoid gradient disappearance and gradient explosion for the back propagation between any two layers, due to the dential mappings of the  $h$  with  $x_{l+1} = y_l$ ,  $y_l = h(x_l) + F(x_l, w_l)$  and  $h(x_l) = x_l$ . If the first term in Eq(5) is made to be a scale factor greater than or less than 1, the gradient may explode or disappear after the multiplication of several back propagations. The more layers, the more obvious it

is. This is also the reason why ResNet performs better than highway network. It should be noted that the BN layer solves the gradient disappearance and explosion of plain net, and the “1” here can avoid the gradient disappearance and explosion on the short cut path.

**B. The structure parameters of FHNet**

After the four residue blocks, the output data goes through another convolution layer, and after that outputs a  $64 \times 64$  transformation matrix  $H$ . The transformation matrix  $H$  is the target matrix and is up sampled to  $512 \times 512$ . Utilizing the transformation matrix  $H$ , the final processed image is got with Eq(3).

The parameters of the Network are given in the Tab.1. Let the size of the input image be  $256 \times 256$ . The input image is a color map has three channels, that are red, blue and green channels. The first convolution layer of FHNet, e.g. Conv\_1, has 16 channels or convolution kernels, and the down-sampling in this layer is performed with a stride of 1 so that the size of the output data keeps unchanged.

The Res\_1 layer has 32 channels with the down-sampling stride be 2. So, the size of the output data from the layer Res\_1 is reduced to be  $256 \times 256$ . The number of the following layers Res\_2, Res\_3 and Res\_4 are 32, 64 and 64 respectively, and the down-sampling stride are 1, 2 and 1. Thus, after the four Residue blocks, the size of the trained data is changed to  $128 \times 128$ .

TABLE I

THE STRUCTURE PARAMETERS OF THE FHNET

Layer name	Kernel size	Stride	Channel	Out Size
Input	-	-	3	$512 \times 512$
Conv_1	$3 \times 3$	1	16	$512 \times 512$
Res_1	$3 \times 3$	2	32	$256 \times 256$
Res_2	$3 \times 3$	1	32	$256 \times 256$
Res_3	$3 \times 3$	2	64	$128 \times 128$
Res_4	$3 \times 3$	1	64	$128 \times 128$
Conv_2	$1 \times 1$	2	3	$64 \times 64$

The second convolution layer Conv\_2 receives the output data from the training Network, reshape it into the three red, blue and green channels again. The Conv\_2 layer performs again a stride 2 convolution for the feature combination and channel number reduction. The Conv\_2 also reduce the size of the input data from  $128 \times 128$  to  $64 \times 64$ . Finally, the

size of feature map, e.g. the transformation matrix H, is  $64 \times 64$ .

It deserves to note that the goal of FHNet is to learn the transformation process of image enhancement, rather not the effect of image enhancement itself. The final output of FHNet is the low resolution transformation matrix H ( $64 \times 64$ ). In the process of inference, one only need to up sample the transformation matrix H to the resolution of the original image, and then transform it to achieve the effect of image enhancement. For example, if the original image resolution is  $512 \times 512$ , if it is directly input into the network, it will easily lead to the problem of memory exhaustion. Therefore, it is

necessary to sample  $256 \times 256$  or even  $96 \times 96$  resolution map before input into the network. In this way, the network only needs to output a low-resolution transformation matrix H. With the transformation matrix H, the image enhancement is achieved. Compared with the image enhancement with the various conventional neural networks which train and learn the image itself, our FHNet is faster and more effective than the neural network with direct output of enhanced images.

### C. Data Set

We use the DPED (DSLR photo enhancement dataset) data set, of which the photos captured by the smartphones are used as the input and the photos taken by the SLR are used as the target. We take the photos with three different smartphones and a digital SLR camera. These photos are taken synchronously in the same field. Devices we have used to collect data include iPhone 11, Huawei P30Pro, XiaoMi 10Pro and Canon 70D DSLR. To ensure that all devices take photos at the same time, they are mounted on a tripod and remotely photographed through a wireless control system.

In one week, more than 20000 photos were collected with these devices, including 4000 from iPhone 11, 5000 from Huawei P30 Pro, 5000 from Huawei and 4400 from Canon 70D DSLR; for each smartphone photo, there was a corresponding photo from Canon SLR camera. These photos were taken during the day in various places, under different light and weather conditions. Images are taken in automatic mode, and we use default settings for all cameras throughout the acquisition process.

The photos, though synchronously captured, are not perfectly aligned as the cameras are with different viewing angles, focal lengths and placed in different positions. To solve this problem, additional non-linear transformations (NLT) is performed with SIFT features. NLT extracts the intersection part between phone and DSLR photos, and then take use of the obtained aligned image fragments to extract patches of desired size for CNN training.

### D. Loss Function

We use the mean square error (L2 loss) as the loss function for the model training

$$MSE = \frac{1}{n} \sum_{i=1}^n (y_i - y_i^p)^2 \quad (6)$$

where  $y_i$  is the input data, and  $y_i^p$  is the expected data.

### E. Training procedure

DPED data is used as the training data of this model. The original image of DPED (such as iPhone 11, Huawei P30 Pro and Xiaomi 10 Pro) is input, and the target image is the enhancement effect image of SLR camera. In essence, FHNet here is the transformation process of learning from ordinary photos to the enhancement image of SLR camera.

In training strategy, we use Adam optimization algorithm, the initial learning rate is set to 0.001, and the learning rate is reduced once in 30, 60 and 120 cycles (epoch)\_ decay= $5e-4$ . The variation of the learning rate with the iteration steps is shown in Fig.3, where it can be seen that only after 20 learning steps, the learning rate can be lowered to  $1e-3$ , indicating that the learning efficiency of our proposed method can be very fast.

Fig.4 and Fig.5 show the training curve of PSNR and L2 loss respectively. It can be seen from Fig.4 and Fig.5 that after 40 cycles of the next iteration, the model starts to stabilize, and the final model convergence can reach 21.8, and L2 loss is less than 0.12.

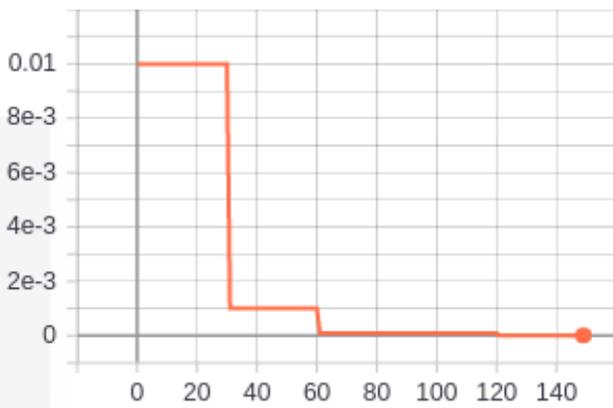


Fig.4 The adjustment strategy of Learning rate

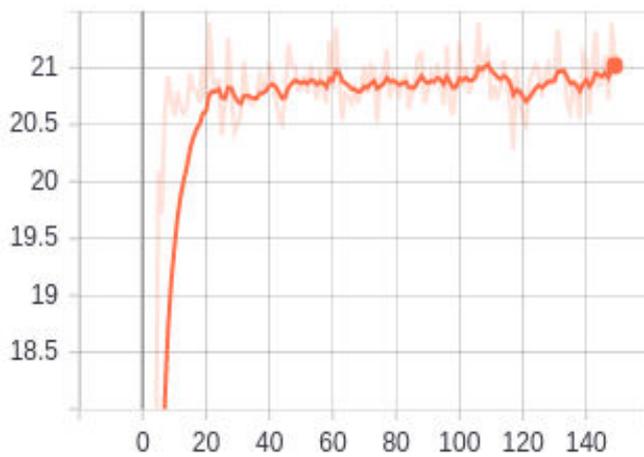


Fig.4 The PSNR curve

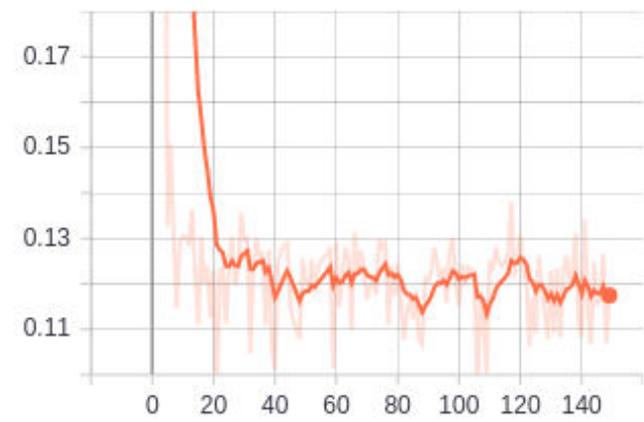


Fig.5. The loss

#### IV. PERFORMANCE AND RESULTS



Fig.6. Original image of the size 512x512



Fig.7. Left is the input image of the size of 256x256 down-sampled from the original image; right is the image of the transformation matrix H that is up-sampled from 64x64 to 256x256.



Fig.8. The enhanced image with FHNet.

Fig.6 depicts the original image with the size to be  $512 \times 512$  that captured with Huawei P30 Pro. The left picture in Fig.7 is the input image of the size of  $256 \times 256$ , down-sampled from the original image(Fig.6), and the right is the image of the transformation matrix which is up-sampled from  $64 \times 64$  to  $256 \times 256$ . Fig.8 is the comparison of the original image with the enhanced image. From Fig.8, it can be seen that, in the image, the leaves of *Dracaena Fragrans* are significantly enhanced compared with other part of the image, indicating that our proposed image enhancement is very effective.

The characteristics of our proposed image enhancement lie in three aspects:

(1) Learning the transformation process, rather than learning the effect of image enhancement: most existed Gan networks directly output the effect map, so they learn the effect of image enhancement directly. In this scheme, the FHNet is constructed to train the transformation matrix  $H$ . What the model learns is the transformation process of image enhancement, not the effect of image enhancement. And, the final enhancement effect is completed by a simple linear transformation, that is,  $G = H \cdot F$ .

Besides, the transformation matrix  $H$  can be learned from the low-resolution image. In the process of reasoning, low resolution transformation matrix  $H$  can also be obtained by low-resolution graph reasoning. For example, if the original image resolution is  $1024 \times 1024$ , if it is directly input into the network, it will easily lead to the problem of memory exhaustion. Therefore, it is necessary to sample  $256 \times 256$  or even  $96 \times 96$  resolution map before input into the network. In this way, the network outputs a low-resolution transformation matrix  $H$ , and then applies the transformation matrix  $H$  to the resolution of the original image, and then performs linear transformation:  $G = H \cdot F$ .

(2) Effectively solve the problem of color distortion: it is easy to cause image distortion by directly up sampling (scaling) the original image. In this scheme, the low-resolution transform matrix  $H$  is to be up sampled to the size of the original image, and then transformed with the original image. Scaling the transform matrix  $H$  will not affect the texture details of the original image. Therefore, this scheme can effectively avoid the problem of image distortion.

(3) Small memory consumption, low computation cost and high speed: the biggest problem of full convolution network is that it takes up more memory. Most of Gan network is trained on low rate map, and then reasoning and testing in full resolution map.

Therefore, in the process of reasoning, Gan network has the disadvantages of large memory and slow speed. FHNet can be trained in low-resolution graph, and can also be used for reasoning and testing in low-resolution graph, so as to reduce the amount of calculation and memory consumption. Therefore, it can effectively solve the problem of high memory and slow speed in the reasoning process of Gan network. At present, GPU test on computer can process  $> 50$ FPS in real time, and in Android test, it can reach the speed of 15-25fps.

## V. CONCLUSIONS

In conclusion, the proposed image enhancement method based on FHNet to train the transformation function is effective for the feature enhancement of the image. FHNet utilizes four residue blocks of the ResNet-18 as the basenet to train the transformation function or matrix of the image. With the learned information contained in the transformation matrix, the enhancement of the image feature is easily implemented with multiplying the transformation matrix with the input images, which can rapidly return a enhanced image with high quality.

Compared with conventional method, this proposed method possess the following charactersitics:(1) Learning the transformation process, not learning the effect of image enhancement;(2) Effectively solve the problem of color distortion;(3) Small memory consumption, low computation cost and high speed.

## ACKNOWLEDGMENT

We appreciate the help from the Second Senior High School of Sanming City, Hujian Province, and the Qiuzhen college, Huzhou University.

## REFERENCES

- [1] K.Pushpanathan , M. Hanafi, S. Mashohor, et al. "Machine learning in medicinal plants recognition: a review." Artificial Intelligence Review:1-23 ,2020.
- [2] You Chenyu, Yang Qingsong, Shan Hongming, Gjestebj Lars, Li Guang , et al. "Structurally-sensitive Multi-scale Deep Neural Network for Low-Dose CT Denoising." IEEE Access PP:1-1,2018.
- [3] Eberhard, Matthias , and H. Alkadhi . "Machine Learning and Deep Neural Networks: Applications in Patient and Scan Preparation, Contrast Medium, and Radiation Dose Optimization." Journal of Thoracic Imaging 35 Suppl 1:1,2020.
- [4] B. Laurent Dercle A , et al. "Using a single abdominal computed tomography image to differentiate five contrast-enhancement phases: A machine-learning algorithm for radiomics-based precision medicine." European Journal of Radiology 125.
- [5] V. Tadic, A. Odry, E. Burkus, I. Kecskes, Z. Kiraly, P. Odry, "Edge-preserving Filtering and Fuzzy Image Enhancement in Depth Images Captured by Realsense Cameras in Robotic Applications," Advances in Electrical and Computer Engineering, vol.20, no.3, pp.83-92,2020.
- [6] Kim, Hong Kyu , et al. "Nanoscale light element identification using machine learning aided STEM-EDS." entific Reports 10.1:13699,2020.

- [7] P. Karthika, P. Vidhya Saraswathi, "IoT using machine learning security enhancement in video steganography allocation for Raspberry Pi." *Journal of Ambient Intelligence and Humanized Computing* 10, 2020.
- [8] V. Tadic, A. Odry, E. Burkus, I. Kecskes, Z. Kiraly, P. Odry, "Edge-preserving Filtering and Fuzzy Image Enhancement in Depth Images Captured by Realsense Cameras in Robotic Applications," *Advances in Electrical and Computer Engineering*, vol.20, no.3, pp.83-92, 2020.
- [9] D. Alis, A. Guler, M. Yergin, O. Asmakutlu, "Assessment of ventricular tachyarrhythmia in patients with hypertrophic cardiomyopathy with machine learning-based texture analysis of late gadolinium enhancement cardiac MRI." *Diagnostic and interventional imaging* 101.3, 2019.
- [10] Saha, Ashirbani, Grimm, Lars J., Ghatge, Sujata V., Kim, Connie E., Soo, Mary S., Yoon, Sora C., Mazurkowski, Maciej A. I., "Machine learning-based prediction of future breast cancer using algorithmically measured background parenchymal enhancement on high-risk screening MRI." *Journal of Magnetic Resonance Imaging*, 2019.
- [11] Zabihollahy, Fatemeh, et al. "Fully automated segmentation of left ventricular scar from 3D late gadolinium enhancement magnetic resonance imaging using a cascaded multi-planar U-Net (CMPU-Net)." *Medical Physics* 47, 2020.
- [12] Inoue, M.; Freel, T.; Van Avermaete, A.; Leevy, W.M. Color Enhancement Strategies for 3D Printing of X-ray Computed Tomography Bone Data for Advanced Anatomy Teaching Models. *Appl. Sci.* 10, 1571, 2020.
- [13] Vanesa Lopez-Vazquez, Jose Manuel Lopez-Guede, Simone Marini, Emanuela Fanelli, Jacopo Aguzzi, "Video Image Enhancement and Machine Learning Pipeline for Underwater Animal Detection and Classification at Cabled Observatories." *Sensors*, 20.3:726, 2020.
- [14] Kevin de Haan, Yair Rivenson, Yichen Wu, Aydogan Ozcan Haan, "Deep-Learning-Based Image Reconstruction and Enhancement in Optical Microscopy." *Proceedings of the IEEE* 108.1:30-50, 2019.
- [15] Fayaz Ali Dharejo, Yuanchun Zhou, Farah Deeba, Yi Du, "A Color Enhancement Scene Estimation Approach for Single Image Haze Removal." *IEEE Geoscience and Remote Sensing Letters* PP.99:1-5, 2019.
- [16] Guoliang Li, Mohammad N.A. Rana, Jinhong Sun, Yinglei Song, Junfeng Qu, "Real-time image enhancement with efficient dynamic programming." *Multimedia Tools and Applications* 1:1-21, 2019.
- [17] S. Jayasankari, S. Dominic, "Contrast Enhancement Using Inverted Gaussian Histogram Specification Technique." *Circuits Systems and Signal Processing* 6, 2020.
- [18] Zhenhong Jia, Li Zhi, Nikola Kasabov, Jie Yang, "Low Illumination Video Image Enhancement." *IEEE Photonics Journal* PP.99:1-1, 2020.
- [19] Jiang, Lincheng, et al. "Deep Refinement Network for Natural Low-Light Image Enhancement in Symmetric Pathways." *Symmetry* 10.10, 2018.
- [20] Pizer, S.M.; Amburn, E.P.; Austin, J.D.; Cromartie, R.; Geselowitz, A.; Greer, T.; ter Haar Romeny, B.; Zimmerman, J.B.; Zuiderveld, K. Adaptive histogram equalization and its variations. *Comput. Vis. Gr. Image Process.* 39, 355–368, 1987.
- [21] Yamasaki, A.; Takauji, H.; Kaneko, S.I.; Kanade, T.; Ohki, H. Denighting: Enhancement of nighttime images for a surveillance camera. In *Proceedings of the 2008 IEEE Conference on Computer Vision and Pattern Recognition*, Tampa, FL, USA, 8–11, pp. 1–4, Dec. 2008.
- [22] Wang, Dongsheng, X. Niu, and Y. Dou. "A piecewise-based contrast enhancement framework for low lighting video." *International Conference on Security IEEE*, 2014.
- [23] Polap, D.; Winnicka, A.; Serwata, K.; Kulesik, K.; Woźniak, M. An Intelligent System for Monitoring Skin Diseases. *Sensors*, 18, 2552, 2018.
- [24] Wozniak, M.; Polap, D. Adaptive neuro-heuristic hybrid model for fruit peel defects detection. *Neural Netw.* 98, 16–33, 2018.
- [25] Wozniak, M.; Polap, D.; Capizzi, G.; Sciuto, G.L.; Kosmider, L.; Frankiewicz, K. Small lung nodules detection based on local variance analysis and probabilistic neural network. *Comput. Methods Progr. Biomed.* 161, 173–180, 2018.
- [26] Chen, L.C.; Papandreou, G.; Kokkinos, I.; Murphy, K.; Yuille, A.L. DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs. *IEEE Trans. Pattern Anal. Mach. Intell.* 40, 834–848, 2018.
- [27] Tadeusiewicz, R. Neural networks in mining sciences-general overview and some representative examples. *Arch. Min. Sci.* 60, 971–984, 2015.
- [28] Gajewski, J.; Vališ, D. The determination of combustion engine condition and reliability using oil analysis by MLP and RBF neural networks. *Tribol. Int.* 115, 557–572, 2017.
- [29] Mustafa Petek, and Bekir Karlik. "Determination of the Mutagenic Effects of Pollution by AMES and Neural Networks." Springer Netherlands, 2009.
- [30] Li, Kuan, Jin, Yi, Akram, Muhammad Waqar, Han, Ruize, Chen, Jiongwei. "Facial expression recognition with convolutional neural networks via a new face cropping and rotation strategy." *The Visual Computer* 36.2:391-404, 2020.
- [31] Li, C.; Min, X.; Sun, S.; Lin, W.; Tang, Z. Deepgait: A learning deep convolutional representation for view-invariant gait recognition using joint bayesian. *Appl. Sci.* 7, 210, 2017.
- [32] Ren, Y.; Zhu, C.; Xiao, S. Small Object Detection in Optical Remote Sensing Images via Modified Faster R-CNN. *Appl. Sci.* 8, 813, 2018.
- [33] Pan, Z.; Jiang, G.; Jiang, H.; Yu, M.; Chen, F.; Zhang, Q. Stereoscopic Image Super-Resolution Method with View Incorporation and Convolutional Neural Networks. *Appl. Sci.* 7, 526, 2017.
- [34] Zhang, K.; Zuo, W.; Chen, Y.; Meng, D.; Zhang, L. Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising. *IEEE Trans. Image Process.* 26, 3142–3155, 2017.
- [35] He, Kaiming, et al. "Deep Residual Learning for Image Recognition." *IEEE Conference on Computer Vision & Pattern Recognition IEEE Computer Society*, 2016.
- [36] Zheng, Yuxuan, et al. "Deep Residual Learning for Boosting the Accuracy of Hyperspectral Pansharpening." *IEEE Geoscience and Remote Sensing Letters* PP.99:1-5, 2019.
- [37] K. Simonyan and A. Zisserman. "Very deep convolutional networks for large-scale image recognition". In *ICLR*, 2015.

**Mail your Manuscript to**  
**[editorijctjournal@gmail.com](mailto:editorijctjournal@gmail.com)**  
**[editor@ijctjournal.org](mailto:editor@ijctjournal.org)**