



Improvement for Single Image Super-resolution and Image Segmentation by Graph Laplacian Regularizer based on Differences of Neighboring Pixels

Lukman Hakim¹ Huipeng Zheng¹ Takio Kurita^{2*}

¹*Department of Information Engineering, Hiroshima University, Japan*

²*Graduate School of Advanced Science and Engineering, Hiroshima University, Japan*

* Corresponding author's Email: tkurita@hiroshima-u.ac.jp

Abstract: The neighboring relationship of the pixels is fundamental information in the image. Utilizing neighboring pixel information provides an advantage over using only pixel-to-pixel information. This study presents a regularization term based on the differences of neighboring pixels. We define the differences of neighboring pixels by using the Graph Theory approach. We then propose an effective graph Laplacian regularization to enforce the differences between pixels as the nodes and differences between nodes as edges. In our scheme, Graph Laplacian constructs from the output of the convolutional neural network and ground-truth image. The generated differences matrices from the output and ground-truth are combined as a Laplacian regularization term used as the deep convolutional neural network's new objective function. Experiments show that our scheme successfully captures pixel neighbor relations and improves the performance of the convolutional neural network better than the baseline without a regularization term. Qualitative and quantitative results show that our developed regularizer proved to enhance the boundary structure on image super-resolution and image segmentation tasks, which is achieved Structural Similarity Index (SSIM) of 0.9604 and 0.8263 on Cartoon set and Manga109 datasets and area under the curve (AUC) of 0.9740 and 0.9561 on DRIVE and STARE datasets, respectively.

Keywords: Graph laplacian regularization, Differences, Neighboring pixel, Boundary structure, Deep learning, Super-resolution, Segmentation.

1. Introduction

Deep learning has greatly impacted computer vision with multiple practical applications such as super-resolution and image segmentation. Deep learning has advantages to learning features from data and is more robust to appearance variations.

For training purpose, many deep learning techniques have focused on deeper architectures while still using a pixel-wise loss, such as Binary Cross-Entropy (BCE) or Mean Square Error (MSE) which is loss calculated based on pixel-to-pixel on predicted output and ground-truth. BCE is often used in image segmentation tasks [1, 2], while MSE is often used in super-resolution tasks [3-5].

However, learning features by using pixel-wise loss fails to preserve the pixel's neighboring relationship in images [6, 7]. We found that pixel-

wise loss is difficult to deal with structure boundary issues. Ideally, the boundary between different classes is clear. However, as shown in Fig. 1, the Convolutional Neural Network (CNN) baseline output tends blurred in the boundary area. In the segmentation task, this leads to the prediction error at the structure boundary as a false positive, leading to errors in the evaluation process. In super-resolution task, it will make failed to generate detailed parts of images. The neighboring relationship of a pixel contains rich information about the spatial structure, local context, and structural knowledge. Preserve the pixel relationship information is very important and generally needed to make network learning more robust.

Pixels neighborhood relationship approach was recently studied to solve the above problem. Zhou [8] presented an affinity space for semantic segmentation

that highlights structure using co-occurring output patterns between neighboring pixels. Ke [6] introduces the concept of Adaptive Affinity Fields (AAF) to preserve and fit the relationships between neighboring pixels in the label space. In [9], the author proposed a regularizer based on the Euler characteristic defined from neighboring pixels to regularize the number of isolated objects based on the network outputs.

Recently, several methods based on the graph theory approach have also been used to learn the pixel relationships information to its neighbors [10-13]. The relationship information between neighboring pixels is captured from the image by being represented as a graph, where a pixel represents a node, and the intensity similarity between pixels is represented as a weighted edge between two nodes. However, all previous study constructs graphs and defines weighted edges only from predicted images while ignoring ground truth in defining graphs. Groundtruth is very important to be involved in defining a graph for an accurate learning process.

In this paper, we propose Graph Laplacian Regularization based on the Differences of Neighboring Pixels (GLRDN) by constructing graph laplacian from both prediction image and ground-truth image. The graphs use the pixel as a node and the edge defined by the “differences” of a neighboring pixel instead of the similarity between pixels. The basic idea is, if pair-wise pixels belonging a similar class, the differences are small. Otherwise, the differences are big if pair-wise pixels belonging a different class.

Then, CNN uses a ground-truth image as a reference to make the differences of the prediction image more similar to the ground-truth image by minimizing the differences between the two images. Specifically, by using the differences of neighboring pixels as a Laplacian regularization term into a deep learning framework, we are successfully capturing pixel relationships information than pure CNN-based approaches.

Our main contributions in this study can be summarized as follows:

- Introduce a novel graph Laplacian regularization term based on differences of neighboring pixels for the CNN model that can capture the neighboring pixels relationships information.
- Construct graph laplacian deriving by two images, from output prediction and ground-truth images.
- Analyzed that adding our GLRDN loss improves the results and enhances the boundary structure on different tasks.

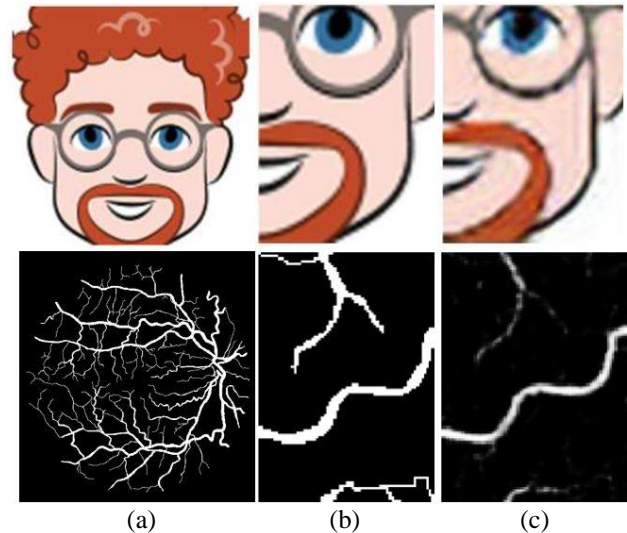


Figure. 1 Example of unclear problem in the boundary area. The first row is an example of an image super-resolution problem. The second row is an example of an image segmentation problem. (a) ground truth in full size, (b) ground truth in patch, and (c) baseline output

This paper is organized as follows: Section 2 describes the previous work related to this study. Section 3 provides details of the proposed methods. Section 4 details the datasets, experimental setup, and performance measures. Section 5 presents the results and discussions. Then, Section 6 provides conclusions, limitations, and future works.

2. Related works

Regularization allows for efficient fine-tuning of network layers, resulting in improved efficiency. This is consistent with previous research, which found that using semantic regularization in a deep network improves accuracy and convergence speed [14].

Regularization based on graphs is used in many applications in the literature. Zeng [15] employs graph Laplacian regularization into a deep learning architecture for real image noise removal. On a patch-by-patch basis, the graphs are built from the CNN outputs. Dinesh [16] proposed a signal-dependent feature graph Laplacian regularizer (SDFGLR) for denoising imperfection 3D point cloud due to the acquisition process means that point clouds are often corrupted with noise. The author assumes that the normal surfaces calculated from point coordinates are smooth in parts with regard to the signal-dependent graph Laplacian matrix. However, the studies need to modify the network architecture to deploy graph Laplacian as a layer. Changing the architecture makes the complexity increase. Therefore, this study only adds the proposed methods on the last part of baseline CNN without changing the architecture.

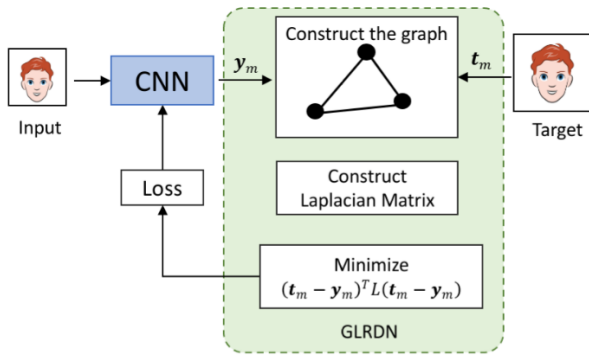


Figure. 2 Visualization of our proposed approach, the graphs constructed from prediction image and ground-truth image. Then, the differences metrics between prediction image and ground-truth image is uses as graph Laplacian regularization term

Ando [17] introduced generalization limitations to learning graphs utilizing the characteristics of the graph in Laplacian regularization. This study showed, in particular, the relevance of laplacian normalization and a decrease in graphic design dimensions. The author stated that the typical L-scaling standard procedure is unsatisfactory because normalization factors might vary substantially throughout a pure component.

In the segmentation task, Li [18] applies graph convolution into the semantic segmentation task use Laplacian to perform reasoning directly to the feature space. Lu [19] generates a neighborhood graph that shows the relationship for each point's neighboring points and then filters the neighborhood graph using Chebyshev polynomials. Hakim [11] introduced a graph Laplacian regularizer, which divides the image into two regions by measuring graph laplacians on the vessels and their backgrounds.

Regularization based on the graph is also used in the Super-resolution task. Xu [20] proposed the graph Laplacian regularization to preserve the spectral information in Hyperspectral images. Liu [12] employs graph Laplacian regularizer to capture the essential piecewise smooth characteristic of the depth map, which has desirable filtering properties. In Hu [21], utilize graph Laplacian regularization for generalized piecewise-smooth image denoising and super-resolution. However, the studies mentioned above are not considering the ground truth image in defining the graph. The graph is only derived from input images or predicted images.

In contrast with existing methods, in this work, we construct the graph laplacian deriving by two images: from prediction and ground-truth, which the adjacency matrix defined by differences of the neighboring pixel instead of the similarity between pixels.

3. Proposed methods

This section introduces graph theory and then presenting our work on a Graph Laplacian Regularizer based on Pixel Differences (GLRDN). Generally, our proposed approach is shown in the Fig. 2.

3.1 Graph laplacian based on differences of neighboring pixels

Graph theory has a long history in mathematics and has been used in many fields of science and engineering. A graph is a representation of the set of elements and the set of relationships between these elements. These elements are called nodes or vertices, and their relations are called edges [22]. In this paper, the graph is used to modeling every pixel in image data as the node and use neighboring pixel information as edge structure for each node.

More formally, let $G = (V, E)$ be a pixel adjacency graph where $V = \{i | i = 1, \dots, N\}$ is the set of the pixel indices which represented as nodes with N pixels and the $E = \{(i, j) | i, j \in V\}$ is the neighboring relations between the pixels which representation of edges. Then the adjacency matrix of this pixel adjacency graph is defined as $A = [a_{ij}]$ where the (i, j) element of the matrix is given as

$$a_{ij} = \begin{cases} 1, & \text{if } (i, j) \in E \\ 0, & \text{otherwise} \end{cases} \quad (1)$$

The degree matrix of this graph is defined as the diagonal matrix $D = \text{diag}(d_j = \sum_{i=1}^N a_{ij})$. Then we can define the graph Laplacian matrix as $L = D - A$.

Let us consider the set of training samples $X = \{(x_m, t_m) | m = 1, \dots, M\}$ where x_m is a m^{th} input image and t_m is the m^{th} target image. The number of training samples is denoted by M .

In deep convolutional neural network, the network is trained to predict the output image y_m from the m^{th} input image x_m .

To define the GLRDN, we consider to utilize the differences of the differences of neighboring pixels between the target image t_m and the estimated images y_m as

$$\begin{aligned} S_G(t_m, y_m) &= \sum_{(i,j) \in E} \{(t_{mi} - t_{mj}) - (y_{mi} - y_{mj})\}^2 \\ &= \sum_{(i,j) \in E} (\Delta t_m^{ij} - \Delta y_m^{ij})^2 \\ &= (\Delta t_m - \Delta y_m)^T (\Delta t_m - \Delta y_m) \\ &= (B t_m - B y_m)^T (B t_m - B y_m) \\ &= (t_m - y_m)^T B^T B (t_m - y_m) \\ &= (t_m - y_m)^T L (t_m - y_m) \end{aligned} \quad (2)$$

where B is the incident matrix.

It is noticed that $B^T B$ is equal to the graph Laplacian matrix L . For example, the incident matrix B and the graph Laplacian matrix L for the simple graph in the Fig. 3(a) are given as Fig. 3(b) and 3(c). Graph Laplacian matrix assumes that the differences of neighbor pixel between target images and estimated images, denoted as $(t_m - y_m)$ is smooth with respect to concerning corresponding graph G . In particular, it enforces the value of $(t_m - y_m)^T L(t_m - y_m)$ should be small.

For M training samples, we can define the graph Laplacian regularization term as

$$S_G = \sum_{m=1}^M S_G(t_m, y_m) = \sum_{m=1}^M (t_m - y_m)^T L(t_m - y_m) \quad (3)$$

This measure S_G becomes small if the estimated output images are similar with the target images.

3.2 The GLRDN with objective function

The neighboring pixel relationship is fundamental information in an image. Any task that involves images in the learning process, especially computer vision, utilizing neighboring pixel information provides an advantage over using only pixel-to-pixel information by adding the GLRDN as a regularization term into pixel-wise loss function.

In this section, we explain how to utilize the GLRDN combining with pixel-wise loss functions such as MSE and BCE. MSE is commonly used in image super-resolution tasks. The MSE is given by

$$E_{mse} = \frac{1}{M} \sum_{m=1}^M (t_m - y_m)^2 \quad (4)$$

For the training of the parameters of the network, we combine the MSE loss with the regularization term as

$$Q = E_{mse} + \lambda S_G \quad (5)$$

where λ is a scaling parameter to control the effect of the regularization.

In the image segmentation task, the binary cross-entropy (BCE) loss is used as the objective function. The BCE loss is given by

$$E_{bce} = -\sum_{m=1}^M \sum_{i=1}^N \{t_{mi} \log(y_{mi}) + (1 - t_{mi}) \log(1 - y_{mi})\} \quad (6)$$

Similarly, we combine the objective function with the proposed regularization term for M training samples as

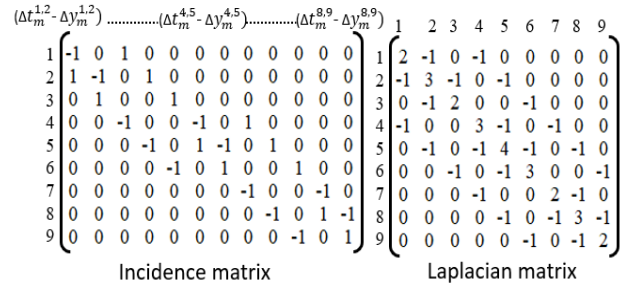
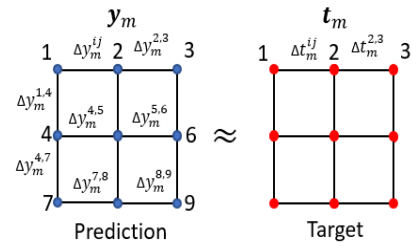


Figure. 3 Differences of neighboring pixels calculation. The top figure shows the basic idea to measure the differences between pixels. The bottom figure shows the Incident matrix and Laplacian matrix calculations example from the incident matrix.

$$Q = E_{bce} + \lambda S_G \quad (7)$$

where λ is a scaling parameter to control the effect of the regularization.

4. Experiments

To know the effectiveness of our proposed approach, we evaluated the proposed approach in two different tasks in computer vision: Images Super-Resolution and Image segmentation. We compare the results to a baseline CNN without regularization. In this section, we describe the datasets, parameter setting, experimental setup, and performance measures.

4.1 Datasets

We conduct the experiments on four datasets: Cartoon Set [23], Manga109 [24], DRIVE [25] and STARE [26] datasets. Cartoon Set is a collection of 2D cartoon avatar images that are selected randomly. With a total of 1013 potential combinations, the cartoons are divided into ten artwork categories, four-color categories, and four proportion categories. We started our experiment on this simple dataset at first. Manga109 was collected by the Aizawa Yamasaki Matsui Laboratory from the University of Tokyo. The collection is designed for use in academic study on Japanese manga media processing. We resized all samples from Cartoon Set and Manga109 to 128x128 pixels. We took 1k images for training and 100 images for testing. We use Cartoon set and

Manga109 dataset for evaluation on image super-resolution tasks. The DRIVE dataset is captured by Canon CR5 nonmydriatic 3CCD camera at 45° field of view, comprised of 40 images. The size of images is 768x584 pixels with 8 bits per color channel. The STARE dataset was captured by TopCon TRV-50 fundus camera at 35° field of view, which consisted of 20 images. Each image has a size of 605x700 pixels and has 24 bits per pixel.

4.2 Experiment settings

The experiments were performed on two different tasks, image super-resolution tasks and image segmentation tasks. For the image super-resolution task, we adopt the Enhanced Deep Residual Networks for Single Image Super-Resolution (EDSR) [27] model as our baseline model due to its great performance on image super-resolution tasks. The EDSR extracts a series of feature maps from low-resolution images. The intermediate layer converts these feature maps to high-resolution patch representations. Furthermore, the last part of EDSR produces the final high-resolution images. Finally, we incorporated the GLRDN after the last layer. In all these settings, we compare the performance with and without our regularizer. We used 4000 epochs and set the batch size at 20. The learning rate is set at 0.0001 and is half after 2000 epochs. Furthermore, the proposed method is compared with state-of-the-art methods in image super-resolution tasks such as bicubic, Super-Resolution Convolutional Neural Network (SRCNN) [3], and Wide Activation for Efficient and Accurate Image Super-Resolution (WDSR) [29]. Bicubic is a commonly used algorithm for scaling images and videos. This algorithm works by replicating pixels and approximating the pixels by the nearest neighbor pixel value in a 4x4 pixel (16 pixels) field. SRCNN is a CNN model designed to map low-resolution images to high-resolution images. SRCNN has three main parts: patch extraction and representation which the image will be enlarged using bicubic interpolation; non-linear mapping, which consists of a 1x1 convolutional layer; and the reconstruction part, which will reconstruct vectors into the image. WDSR is the residual network for single image super-resolution, which uses wider features before the activation function. WDSR has a slim identity mapping with a wider channel before activation in each residual block to get a better trade-off between accuracy and efficiency.

For image segmentation, we use U-Net [28] model as our baseline model that contains encoder modules and decoders modules. The encoder path has three units. In the first unit, two convolutional layers

are followed by Rectified Linear Unit (ReLU) and max-pooling layer, with each 32 feature maps. In the second unit, two convolutional layers are followed by the ReLU and max-pooling layer with 32 and 64 feature maps. In the third unit, two convolutional layers with each 128 feature maps. The decoder path has four convolutional layers followed by ReLU and one convolution layer without ReLU. After the decoder path, the output is upsampled with the factor of 2. Lastly, the feature maps were joining with the encoder layers by skip connections. The sigmoid classifier function applies to probability map, and then we added GLRDN after the sigmoid function. We used 200 epochs and set the batch size to 16. The learning rate is set at 0.001 and decreased ten times every 25 epochs to ensure network convergence. We utilized Adam to optimize the network. Experiments have set the regularization parameters λ to $1e-5$. The proposed method is implemented using the Pytorch and train using GPU machine Nvidia GeForce GTX 1080/PCIe/SSE2 graphic card on Intel(R) Core(TM) i7-6700K CPU @ 4.00GHz Processor, 32 GB of RAM.

The number of training and testing data of the DRIVE dataset used in this study are 20 and 20, respectively. In order to ensure that there are enough training images, we divided the images into patches

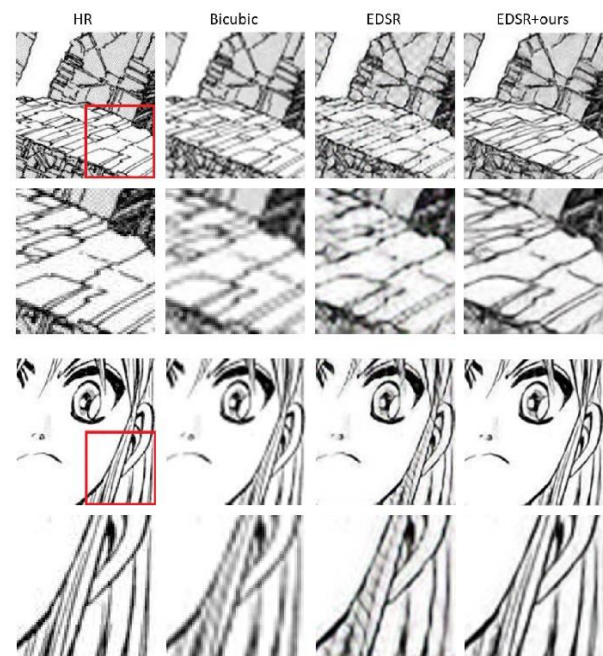


Figure. 4 The results of the architecture without and with GLRDN on the Manga109 dataset. The first and third rows show a comparison in full images. The second and fourth rows show the comparison in zoom mode in the red rectangle area to highlight the differences in more detail

with the size of 48x48 pixels. Hence the total number of patches generated on every image is 4.750 patches. Thus a total of 950.000 patches was generated from the training images to increase the size of the dataset. We also use the same setting for experiments on STARE datasets.

The GLRDN adopts a 4-neighbor strategy. When calculating its difference matrix, this strategy requires a large memory to hold a 4× ground truth label and predicted probabilities. We use two approaches to overcome this problem: firstly, when choosing the neighbor pixels in a square region, we set the region size to be 3×3 as suggested. Secondly, we calculate the GLRDN from n samples for every minibatch randomly. In experiments, we use n=5 sample images for each minibatch. This approach reduces the complexity time significantly.

4.3 Performance measure

We evaluate the improvement of our method in the images super-resolution task using Peak Signal to Noise Ratio (PSNR) and Structure Similarity Index Method (SSIM). PSNR is the ratio calculated between the highest potential signal and distorting noise that influences its representation accuracy. Given a ground-truth image and a predicted image, the PSNR is defined by:

$$PSNR = 10 \log_{10} (255^2 / MSE) \quad (8)$$

SSIM is used for measuring the similarity between two images. The SSIM is built by modeling image distortion as a combination of three factors: lack of correlation, luminance distortion, and contrast distortion. The SSIM is defined as :

$$SSIM = l(x, t)c(x, t)s(x, t) \quad (9)$$

where l , c , and t are luminance distortion, contrast distortion, and lack of correlation, respectively.

For the image segmentation task, we use the area under the curve (AUC), sensitivity (S_n), specificity (S_p), Accuracy (Acc) to measure the performance of our proposed method. Acc calculates the proportion of the predicted vessel pixels, which are true vessel pixels. S_n evaluates the proposed structure's ability to detect vessel pixels. S_p evaluates the ability of the structure to detect background pixels. The metric measurements of Acc , S_n , and S_p are defined as follows :

$$Acc = \frac{TP+TN}{TP+FN+TN+FP} \quad (10)$$

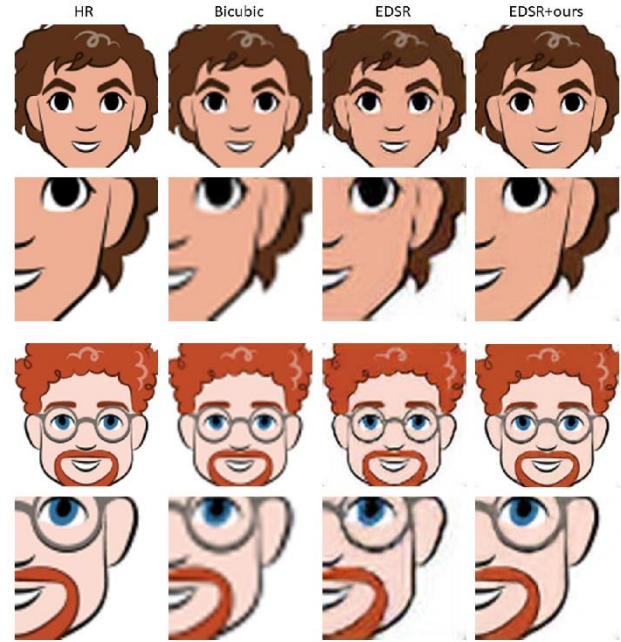


Figure. 5 The results of the architecture without and with GLRDN on Cartoon set. The first and third rows show the comparison in full images, the second and fourth rows showing a comparison in zoom mode to highlight the differences in more detail

$$S_n = \frac{TP}{TP+FN} \quad (11)$$

$$S_p = \frac{TN}{TN+FP} \quad (12)$$

where TP is true positive, which defines the vessel pixel that is properly classified, FN is a false negative which defines the vessel pixel is classified as a non-vessel pixel. TN is a true negative which referred to as a non-vessel pixel that correctly labeled. FP is a false positive, which defines a non-vessel pixel that is classified as a vessel. We also calculated the AUC value using the receiver operating characteristic curve to evaluate the proposed capability to detect vessel pixels. The receiver operating characteristic (ROC) curve is used to determine AUC .

5. Results and discussion

The proposed network with the GLRDN succeeds in reconstructing the boundary structure detail compared to the network without the GLRDN presented in Fig. 4 and Fig. 5. On the Manga109 dataset, the CNN architecture without the GLRDN compared to the original High Resolution (HR) image, the line portion of the bicubic and EDSR results appear jagged and not smooth. Coupling EDSR with GLRDN, the boundary structure is better

Table 1. Comparison of architectural performance without and with our proposed GLRDN on the super image resolution task. Performance was measured using PSNR and SSIM on Cartoon Set and Manga109 datasets

| Datasets | Methods | Scale | PSNR | SSIM |
|-------------|------------|-------|--------------|---------------|
| Cartoon Set | EDSR | x2 | 25.02 | 0.8743 |
| | EDSR+GLRDN | x2 | 30.51 | 0.9604 |
| Manga109 | EDSR | x2 | 20.54 | 0.7791 |
| | EDSR+GLRDN | x2 | 21.89 | 0.8263 |

Table 2. Comparison of architectural performance without and with our proposed GLRDN on the image segmentation task. Performance is measured using Sn, Sp, Acc, and AUC on DRIVE and STARE datasets

| Datasets | Regularizer | Sn | Sp | Acc | AUC |
|----------|---------------|---------------|---------------|---------------|---------------|
| DRIVE | U-Net | 0.7429 | 0.9840 | 0.9544 | 0.9686 |
| | U-Net + GLRDN | 0.7914 | 0.9791 | 0.9561 | 0.9740 |
| STARE | U-Net | 0.7625 | 0.9624 | 0.9326 | 0.9497 |
| | U-Net + GLRDN | 0.8040 | 0.9580 | 0.9350 | 0.9561 |

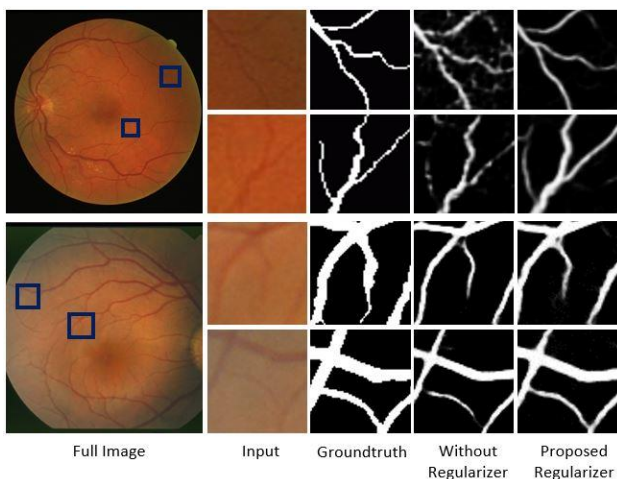


Figure. 2 Comparison results of blood vessel segmentation between our proposed regularizer and without proposed regularizer on the DRIVE (top row) and STARE (bottom row) datasets

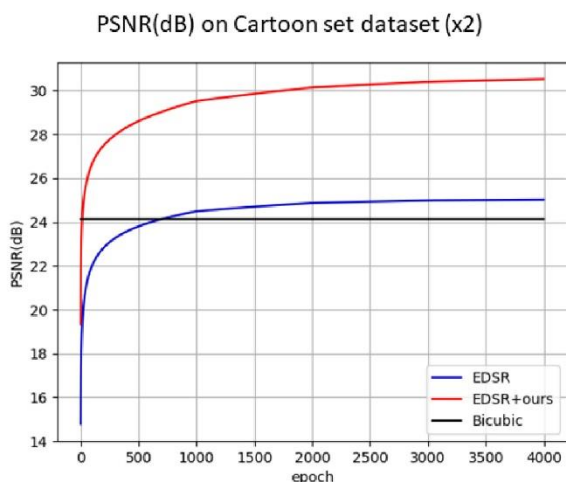


Figure. 3 PSNR comparison of our proposed method on Cartoon set dataset

and the line is smoother close to the HR image. This is because GLRDN considers the difference in the relationship between a pixel and its neighbors in the predicted image and compares it with the difference in the relationship between the pixels in the ground-truth image. Pixels in the boundary area have a significant difference from neighboring pixels. GLRDN helps provide a guideline for the CNN baseline to account for the relationship of different pixels to their neighbors.

On the Cartoon set, the architecture without GLRDN results is not clear. For example, the color is inconsistent in the face skin region that has the same color. Using the proposed network with GLRDN, the face skin color is more consistent and clear. The proposed approach offers smoother line sections and more consistent sections of the same color based on the results. This is because the face skin color area in the cartoon set has the same color, so the relationship between pixels and their neighbors has no difference or small differences. In this case, GLRDN provides the information to the CNN baseline in the learning process so that the prediction ability is better for the face skin color section. Some selected qualitative results on the DRIVE and STARE datasets are shown in Fig. 6. Segmentation results of the proposed network with GLRDN have more accurate connectivity and detail boundary than results without GLRDN. This demonstrates that GLRDN can definitely capture the relationship between pixels in an image by using differences between pixels. Thus, the prediction of the model with GLRDN has better visual qualities. It is because GLRDN considered the differences between neighboring pixels, which means the pixels belonging same classes will force the pixel pairs to make their predictions more

consistent. The neighbor pixels which belong to different classes will separate forces on them to make their predictions more inconsistent.

Table 1 and Table 2 presents the quantitative performance comparison of our proposed approach in image super-resolution and image segmentation task, respectively. The proposed network with GLRDN performed well in capturing the relationship of neighboring pixels better than the network without GLRDN. With EDSR as baseline models, GLRDN can improve the PSNR from 25.02 to 30.51 on the Cartoon set dataset and 20.54 to 21.89 on the Manga109 dataset with upscaling factor of 2. Fig. 7

plotting an example of PSNR visualization on a Cartoon set dataset. In the SSIM measurement, GLDRN improved the result from 0.8743 to 0.9604 on the Cartoon set and 0.7791 to 0.8263 on the Manga109 dataset.

Furthermore, Table 2 showing the performance results on DRIVE and STARE datasets. The GLRDN can improve the AUC score by 0.62 and 0.67 percent on the DRIVE and STARE datasets, respectively. We can also see that the reproduced models with GLRDN show a competitive sensitivity performance by

Table 3. Comparison of the proposed approach performance on image super-resolution task with state-of-the-art methods.

| Methods | Scale | Cartoon set | | Manga109 | |
|-----------|-------|--------------|---------------|--------------|---------------|
| | | PSNR | SSIM | PSNR | SSIM |
| bicubic | x2 | 24.13 | 0.8972 | 19.95 | 0.7887 |
| SRCNN | x2 | 24.15 | 0.8936 | 20.07 | 0.7811 |
| WDSR | x2 | 24.86 | 0.8760 | 20.31 | 0.7758 |
| EDSR | x2 | 25.02 | 0.8743 | 20.54 | 0.7791 |
| EDSR+ours | x2 | 30.51 | 0.9604 | 21.89 | 0.8263 |
| bicubic | x3 | 20.06 | 0.7531 | 16.47 | 0.6120 |
| SRCNN | x3 | 20.41 | 0.7543 | 16.79 | 0.6085 |
| WDSR | x3 | 21.50 | 0.7725 | 17.05 | 0.6188 |
| EDSR | x3 | 21.74 | 0.7737 | 17.15 | 0.6244 |
| EDSR+ours | x3 | 25.96 | 0.8983 | 17.88 | 0.6749 |
| bicubic | x4 | 17.95 | 0.6398 | 14.70 | 0.4915 |
| SRCNN | x4 | 18.47 | 0.6491 | 15.25 | 0.4879 |
| WDSR | x4 | 19.45 | 0.6880 | 15.46 | 0.5056 |
| EDSR | x4 | 19.68 | 0.6793 | 15.53 | 0.5074 |
| EDSR+ours | x4 | 22.66 | 0.8061 | 15.85 | 0.5397 |

Table 4. Comparison of the proposed approach performance on image segmentation with state-of-the-art methods

| Methods | Sp | Sn | Acc | AUC |
|-------------------|--------------|---------------|---------------|---------------|
| Orlando [36] | 0.9684 | 0.7897 | 0.9454 | 0.9506 |
| Hu [38] | 0.9793 | 0.7772 | 0.9533 | 0.9759 |
| Khan [37] | 0.9670 | 0.7373 | 0.9501 | - |
| Fraz [39] | 0.9807 | 0.7406 | 0.9480 | 0.9747 |
| Zhou [31] | 0.9803 | 0.7262 | 0.9475 | - |
| Yan [33] | 0.982 | 0.7631 | 0.9538 | 0.9750 |
| Chen [30] | 0.9735 | 0.7426 | 0.9453 | 0.9516 |
| Zhang [32] | 0.9712 | 0.7861 | 0.9466 | 0.9703 |
| Wang [34] | 0.9736 | 0.7986 | 0.9511 | 0.9740 |
| Strisciuglio [35] | 0.9702 | 0.7777 | 0.9454 | 0.9597 |
| Proposed method | 0.9791 | 0.7914 | 0.9561 | 0.9740 |

6.5 and 5.4 percent on the DRIVE and STARE datasets, respectively. The results demonstrate the effectiveness of the graph Laplacian regularizer based on differences of neighbor pixels.

We also compare the results of the proposed approach with the results of state-of-the-art methods. For a fair comparison, we separate the comparisons according to the task. Table 3 presents the proposed network with GLRDN performance compare with state-of-the-art in image super-resolution tasks which is higher than the other existing methods on all upscaling factors. The proposed network is compared with bicubic, SRCNN, and WDSR. On upscaling factor x2, our proposed approach achieved 30.51 of PSNR and 0.9604 of SSIM on Cartoon set and 21.89 of PSNR and 0.8263 of SSIM. On upscaling factor x3, our proposed regularizer achieved 25.96 of PSNR and 0.8983 of SSIM on Cartoon set and 17.88 of PSNR and 0.6749 of SSIM. On upscaling factor x4, our proposed regularizer achieved 22.66 of PSNR and 0.8061 of SSIM on Cartoon set and 15.85 of PSNR and 0.5397 of SSIM. Compared to the existing approach, our proposed approach is simple and reliable, which only adds one layer after the last layer to enhance the boundary structure area. Meanwhile, other architectures such as WDSR use residual blocks and wider channels on each block to increase accuracy leading to increased complexity.

Table 4 presents comparison of the proposed method performance with state-of-the-art methods on DRIVE datasets. The experiment results of all methods are taken from their published papers. The proposed method got an AUC of 0.9740 and an accuracy of 0.9561, nearly identical to the state-of-the-art method. Although our approach is qualitatively superior to other methods, the ground truth-based quantitative calculations show the same accuracy and AUC value as current methods. This is because the ground truth is that there is no annotation of the small vein, which makes the calculation inaccurate, and the small vessel is considered a false positive. Good detection of small blood vessels contributes to good sensitivity. We can still get very high S_n at a very low S_p price by adjusting the decision thresholds, or vice versa. As a result, AUC is the most important performance predictor of the three, while S_n and S_p are primarily used for detail.

Although our approach proves its ability to enhance the boundary structure on overall datasets, the GLRDN has limitations calculated by treating pixel by pixel, which is a computational time cost. We plan to extend GLRDN by using a sparse graph approach to reduce computational costs in future work.

6. Conclusion

In this paper, we proposed a novel graph approach to construct graph Laplacian regularization term based on differences of neighboring pixels for the deep convolutional neural network.

Compared with baseline architecture regularizer, the proposed approach demonstrated improvements in enhancing the detail of boundary structure on various types of datasets. Furthermore, it also offers the advantage of smoother line sections, and more consistent sections of the same color.

These findings suggest that the proposed system can be very reliable for computer vision tasks. The calculation of GLRDN on a pixel-by-pixel basis is a limitation of this approach. To address this limitation, we plan to expand the work by using sparse graphs to reduce computational costs as future works.

Conflicts of Interest

The authors declare no conflict of interest.

Author Contributions

Conceptualization, Lukman Hakim and Takio Kurita; methodology, Lukman Hakim; software, Lukman Hakim and Huipeng Zheng; validation, Lukman Hakim and Huipeng Zheng; formal analysis, Lukman Hakim and Huipeng Zheng; investigation, Lukman Hakim and Huipeng Zheng; writing—original draft preparation, Lukman Hakim; writing—review and editing, Takio Kurita; visualization, Lukman Hakim and Huipeng Zheng; supervision, Takio Kurita; funding acquisition, Takio Kurita”.

Acknowledgments

This work was partly supported by JSPS KAKENHI Grant Number 21K12049.

References

- [1] J. Long, E. Shelhamer, and T. Darrell, “Fully conVolutional networks for semantic segmentation”, In: *Proc. of 2015 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3431-3440, 2015.
- [2] L. C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, “DeepLab: Semantic Image Segmentation with Deep ConVolutional Nets, Atrous ConVolution, and Fully Connected CRFs”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 40, No. 4, pp. 834-848, 2018.
- [3] C. Dong, C. 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. 295-307, 2016.
- [4] J. Kim, J. K. Lee, and K. M. Lee, “Accurate Image Super-Resolution Using Very Deep Convolutional Networks”, In: *Proc. of 2016 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1646-1654, 2016.
- [5] C. Ledig, L. Theis, F. Huszar, J. Caballero, A. Cunningham, A. Acosta, A. Aitken, A. Tejani, J. Totz, Z. Wang, and W. Shi, “Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network”, In: *Proc. of 2017 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 105-114, 2017.
- [6] T. W. Ke, J. J. Hwang, Z. Liu, and S. X. Yu, “Adaptive Affinity Fields for Semantic Segmentation”, In: *Proc. of the European Conference on Computer Vision*, 2018.
- [7] M. Tang, A. Djelouah, F. Perazzi, Y. Boykov, and C. Schroers, “Normalized Cut Loss for Weakly-Supervised CNN Segmentation”, In: *Proc. of 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1818-1827, 2018.
- [8] W. Zhou, Y. Wang, J. Chu, J. Yang, X. Bai, and Y. Xu, “Affinity Space Adaptation for Semantic Segmentation Across Domains”, *IEEE Transactions on Image Processing*, Vol. 30, pp. 2549-2561, 2021.
- [9] L. Hakim, M. S. Kavitha, N. Yudistira, and T. Kurita, “Regularizer based on Euler characteristic for retinal blood vessel segmentation”, *Pattern Recognition Letters*, Vol. 149, pp. 83-90, 2021.
- [10] L. Zhang, X. Li, A. Arnab, K. Yang, Y. Tong, and P. H. S. Torr, “Dual Graph Convolutional Network for Semantic Segmentation.”, In: *Proc. of the British Machine Vision Conference 2019*, pp. 254 -260, 2019.
- [11] L. Hakim, N. Yudistira, M. S. Kavitha, and T. Kurita, “U-Net with Graph Based Smoothing Regularizer for Small Vessel Segmentation on Fundus Image”, In: *Proc. of International Conference on Neural Information Processing*, pp. 515-522, 2019.
- [12] Y. Liu, S. Yang, B. Li, W. Zhou, J. Xu, H. Li, and Y. Lu, “Affinity Derivation and Graph Merge for Instance Segmentation”, In: *Proc. of the European Conference on Computer Vision*, pp. 708-724, 2018.
- [13] J. Pang and G. Cheung, “Graph Laplacian Regularization for Image Denoising: Analysis in the Continuous Domain”, *IEEE Transactions on Image Processing*, Vol. 26, No. 4, pp. 1770-1785, 2017.
- [14] J. Mojoo, Y. Zhao, M. S. Kavitha, J. Miyao, and T. Kurita, “Completion of Missing Labels for Multi-Label Annotation by a Unified Graph Laplacian Regularization”, *IEICE Transactions on Information and Systems*, Vol. 103, pp. 2154-2161, 2020.
- [15] J. Zeng, J. Pang, W. Sun, and G. Cheung, “Deep Graph Laplacian Regularization for Robust Denoising of Real Images”, In: *Proc. of 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pp. 1759-1768, 2019.
- [16] C. Dinesh, G. Cheung, and I. V. Bajic, “Point Cloud Denoising via Feature Graph Laplacian Regularization”, *IEEE Transactions on Image Processing*, Vol. 29, pp. 4143-4158, 2020.
- [17] R. K. Ando and T. Zhang, “Learning on Graph with Laplacian Regularization”, *Advances in Neural Information Processing Systems 19*, pp. 25-32, 2006.
- [18] X. Li, Y. Yang, Q. Zhao, T. Shen, Z. Lin, and H. Liu, “Spatial pyramid based graph reasoning for semantic segmentation”, In: *Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8950-8959, 2020.
- [19] Q. Lu, C. Chen, W. Xie, and Y. Luo, “PointNGCNN: Deep convolutional networks on 3D point clouds with neighborhood graph filters”, *Computers and Graphics*, Vol. 86, pp. 42-51, 2020.
- [20] Y. Xu, Z. Wu, J. Chanussot, and Z. Wei, “Hyperspectral Images Super-Resolution via Learning High-Order Coupled Tensor Ring Representation”, *IEEE Transactions on Neural Networks*, Vol. 31, No. 11, pp. 4747-4760, 2020.
- [21] W. Hu, G. Cheung, X. Li, and O. C. Au, “Graph-based joint denoising and super-resolution of generalized piecewise smooth images”, In: *Proc. of 2014 IEEE International Conference on Image Processing*, pp. 2056-2060, 2014.
- [22] O. Lézoray and L. Grady, “Graph theory concepts and definitions used in image processing and analysis”, *Image Processing and Analysing With Graphs: Theory and Practice*, pp. 1-24, 2012.
- [23] F. Cole, I. Mosseri, D. Krishnan, A. Sarna, A. Maschinot, B. Freeman, and S. Fuman, “Cartoon Set”, [Online]. Available: <https://google.github.io/cartoonset/>. [Accessed 24 Mei 2021].
- [24] Y. Matsui, K. Ito, Y. Aramaki, A. Fujimoto, T. Ogawa, T. Yamasaki, and K. Aizawa, “Sketch-based manga retrieval using manga109 dataset”,

- Multimedia Tools and Applications*, Vol. 76, pp. 21811-21838, 2017.
- [25] J. Staal, M. Abramoff, M. Niemeijer, M. Viergever, and B. V. Ginneken, "Ridge-based vessel segmentation in color images of the retina", *IEEE Transactions on Medical Imaging*, Vol. 23, No. 4, pp. 501-509, 2004.
- [26] A. Hoover, V. Kouznetsova, and M. Goldbaum, "Locating blood vessels in retinal images by piecewise threshold probing of a matched filter response", *IEEE Transactions on Medical Imaging*, Vol. 19, No. 3, pp. 203-210, 2000.
- [27] B. Lim, S. Son, H. Kim, S. Nah, and K. M. Lee, "Enhanced deep residual networks for single image super-resolution", In: *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pp. 136-144, 2017.
- [28] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional Networks for Biomedical Image Segmentation", In: *Proc. of International Conference on Medical Image Computing and Computer-Assisted Intervention*, pp. 234-241, 2015.
- [29] J. Yu, Y. Fan, J. Yang, N. Xu, Z. Wang, X. Wang, and T. S. Huang, "Wide Activation for Efficient and Accurate Image Super-Resolution.", *ArXiv Preprint ArXiv:1808.08718*, 2018.
- [30] Y. Chen, "A labeling-free approach to supervising deep neural networks for retinal blood vessel segmentation", *ArXiv Preprint ArXiv:1704.07502*, 2017.
- [31] C. Zhou, X. Zhang, and H. Chen, "A new robust method for blood vessel segmentation in retinal fundus images based on weighted line detector and hidden Markov model.", *Computer Methods and Programs in Biomedicine*, Vol. 187, pp. 105231-105231, 2020.
- [32] J. Zhang, Y. Chen, E. Bekkers, M. Wang, B. Dashtbozorg, and B. M. T. H. Romeny, "Retinal vessel delineation using a brain-inspired wavelet transform and random forest", *Pattern Recognition*, Vol. 69, pp. 107-123, 2017.
- [33] Z. Yan, X. Yang, and K. T. Cheng, "A Three-Stage Deep Learning Model for Accurate Retinal Vessel Segmentation", *IEEE Journal of Biomedical and Health Informatics*, Vol. 23, No. 4, pp. 1427-1436, 2019.
- [34] C. Wang, Z. Zhao, Q. Ren, Y. Xu, and Y. Yu, "Dense U-net Based on Patch-Based Learning for Retinal Vessel Segmentation", *Entropy*, Vol. 21, No. 2, p. 168, 2019.
- [35] N. Strisciuglio, G. Azzopardi, M. Vento, and N. Petkov, "Supervised vessel delineation in retinal fundus images with the automatic selection of B-COSFIRE filters", *Machine Vision and Applications Archive*, pp. 1137-1149, 2016.
- [36] J. I. Orlando, E. Prokofyeva, and M. B. Blaschko, "A Discriminatively Trained Fully Connected Conditional Random Field Model for Blood Vessel Segmentation in Fundus Images", *IEEE Transactions on Biomedical Engineering*, Vol. 64, No. 1, pp. 16-27, 2017.
- [37] M. A. U. Khan, T. A. Soomro, T. M. Khan, D. G. Bailey, J. Gao, and N. Mir, "Automatic retinal vessel extraction algorithm based on contrast-sensitive schemes", In: *Proc. of 2016 International Conference on Image and Vision Computing New Zealand*, pp. 1-5, 2016.
- [38] K. Hu, Z. Zhang, X. Niu, Y. Zhang, C. Cao, F. Xiao, and X. Gao, "Retinal vessel segmentation of color fundus images using multiscale convolutional neural network with an improved cross-entropy loss function", *Neurocomputing*, Vol. 309, pp. 179-191, 2018.
- [39] M. M. Fraz, P. Remagnino, A. Hoppe, B. Uyyanonvara, A. R. Rudnicka, C. G. Owen, and S. A. Barman, "An Ensemble Classification-Based Approach Applied to Retinal Blood Vessel Segmentation", *IEEE Transactions on Biomedical Engineering*, Vol. 59, No. 9, pp. 2538-2548, 2012.