# 論 文 の 要 旨

題目　Regularizer based on Pixel Neighboring Relationship for Deep Convolutional Neural Network in Image Segmentation
(画像セグメンテーションにおける深い畳み込みニューラルネットワークのピクセル隣接関係に基づく正則化)

Lukman Hakim

## 1. Introduction

Image segmentation is the technique that divides the image into several regions. As a fundamental part of the image recognition system, image segmentation can be categorized as a pixel classification problem. Each pixel in the image will be classified with a specific label.

Recently, many deep learning techniques have been developed for image segmentation that offers high accuracy and deep architecture such as : Fully Convolutional Network (FCN), U-Net and SegNet . Deep learning has advantages to learning features from data and is more robust to appearance variations.

However, the deep learning technique uses a pixel-wise loss function for the training process like Binary Cross Entropy, Cross Entropy, Mean Square Error and Dice Loss. Using pixel-wise loss neglected the pixel neighbor relationships in the network learning process. In addition, segmentation has the issue of providing a segmentation technique with high accuracy and high efficiency, which is very challenging. Some methods offer the advantage of high accuracy but require high computational costs. Other methods provide high efficiency but with less efficient accuracy.

The neighboring relationship of the pixels is essential information in the image. Utilizing neighboring pixel information provides an advantage over only pixel-to-pixel information and offers lower computational costs than changing the inside of an architecture.

## 2. Proposed Methods

To address the problem mentioned above, we propose several works to including pixel relationships information along pixel-wise loss to neural network. This study presents regularizers to provide the pixel neighbor relationship information to the learning process. We have proposed three regularizers are constructed by the graph theory approach and topology approach: Graph-Based Smoothing Regularizer (GBS), Graph Laplacian Regularization based on the Differences of Neighboring Pixels (GLRDN), and Regularizer based on Euler Characteristic (EC). Figure 1 show the illustration of the general system of all architectures uses in this study. Image as basic information uses as input for CNN,

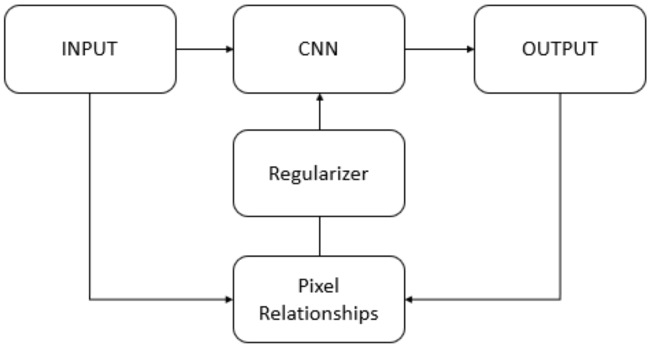


*Figure 1 Illustration of the general system of all architectures uses in this study. Image as basic information uses as input for CNN, and then CNN learns the feature of the image to generate the output. Furthermore, the pixel relationships are extracted from the output and input and used as a regularizer for CNN.*

and then CNN learns the feature of the image to generate the output. Furthermore, the pixel relationships are extracted from the output and input and used as a regularizer for CNN.

## 2.1 Graph Based Smoothing Regularizer

The basic idea of Graph-Based Smoothing Regularizer (GBS) is considers the graph Laplacian from the foreground and background regions and then combines it with the CNN baseline loss function. The combination of regularizers allows the network to learn the pixel relationship efficiently. We formally defined two graphs $G_F$ and $G_B$ for foreground and background, respectively as shown in Fig. 2. $G_F$ and $G_B$ includes a pair of $(V_F, E_F)$, $(V_B, E_B)$, respectively. The parameters $V_F$ and $V_B$ are finite set of elements called vertices, and $(E_F=\{(j_F, k_F) \mid j_F \in V_F, k_F \in V_F\}$ and $(E_B=\{(j_B, k_B) \mid j_B \in V_B, k_B \in V_B\}$ are edges.

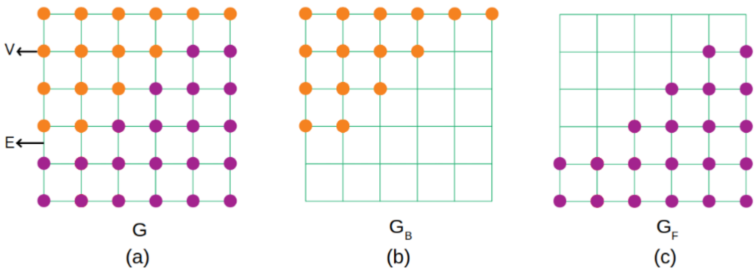


*Figure 2 Representation of set of nodes of the pixel graph to compute boundary between two regions, a) image as an graph, b) background region graph and c) foreground region graph.*

## 2.2 Graph Laplacian Regularization based on the Differences of Neighboring Pixels

Secondly, we introduced Graph Laplacian Regularization based on the Differences of Neighboring Pixels (GLRDN) by constructing graph laplacian from prediction and ground-truth images. A graph uses pixels as vertices and edges defined by the "differences" of neighboring pixels instead of similarities between pixels. A graph uses pixels as vertices and edges defined by the "differences" of neighboring pixels instead of similarities between pixels. The basic idea is shown inf Fig. 3, if pair-wise pixels belonging a similar class, the differences are small. Otherwise, the differences are significant if pair-wise pixels are belonging a different class.

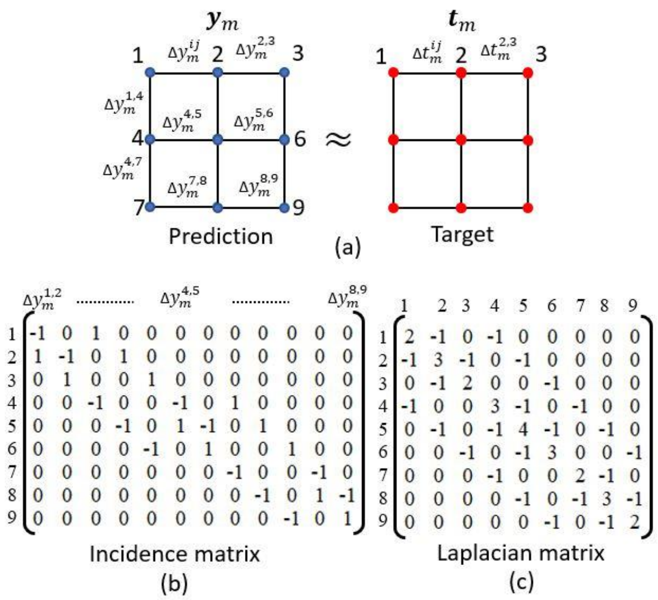


*Figure 3 Differences of neighboring pixels calculation. (a) The basic idea to measure the differences between pixels. (b) Incident matrix calculations example from (a). (c) Laplacian matrix calculated from incident matrix.*

## 2.3 Regularizer based on Euler Characteristic

Thirdly, we proposed a regularizer based on Euler Characteristic (EC) in the segmented image. EC is a topological property of a shape in an image that can identify the number of objects. This EC-based regularizer is used to identify the number of objects in the segmentation results. Then this information is added to the objective function so that the network can minimize the number of isolated objects in the segmented image. Fig. 4 show the example how the number of isolated objects calculated from image.

To evaluate the effectiveness of the proposed methods, we implemented the regularizers on retinal blood vessels segmentation in fundus images and compared them with the baseline CNN without regularizers. 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 and gives a better trade-off between accuracy and efficiency. We offer methods with high accuracy capabilities with less computational costs.

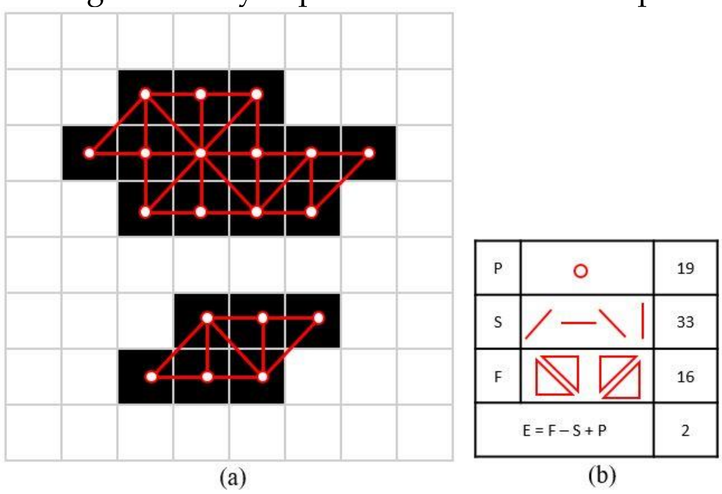


*Figure 4 An example of constructing Euler characteristics using mask patterns (a) illustration showing object (black) pixels based on 8x8 neighborhood pixels. (b) illustration calculating Euler characteristics for the number of isolated object according to the number of vertices (P), sides (S), and faces (F) from object pixels (a).*