Paper:

Self-Organizing Fusion Neural Networks

Jung-Hua Wang, Chun-Shun Tseng, Sih-Yin Shen, and Ya-Yun Jheng

Electrical Engineering Department, National Taiwan Ocean University
2 Peining Rd. Keelung, Taiwan
E-mail: jhwang@mail.ntou.edu.tw
[Received January 16, 2007; accepted March 20, 2007]

This paper presents a self-organizing fusion neural network (SOFNN) effective in performing fast clustering and segmentation. Based on a counteracting learning scheme, SOFNN employs two parameters that together control the training in a counteracting manner to obviate problems of over-segmentation and under-segmentation. In particular, a simultaneous region-based updating strategy is adopted to facilitate an interesting fusion effect useful for identifying regions comprising an object in a self-organizing way. To achieve reliable merging, a dynamic merging criterion based on both intra-regional and inter-regional local statistics is used. Such extension in adjacency not only helps achieve more accurate segmentation results, but also improves input noise tolerance. Through iterating the three phases of simultaneous updating, self-organizing fusion, and extended merging, the training process converges without manual intervention, thereby conveniently obviating the need of pre-specifying the terminating number of objects. Unlike existing methods that sequentially merge regions, all regions in SOFNN can be processed in parallel fashion, thus providing great potentiality for a fully parallel hardware implementation.

Keywords: neural networks, image segmentation, clustering, counteracting learning, watershed

1. Introduction

The goal of image segmentation is to identify objects of interest that satisfy certain pre-defined homogeneity criteria, whereas the primary objective of clustering [1] is to partition a given set of data or objects into clusters (subsets, groups, classes). This partition should have the following properties: 1) *Homogeneity* within the clusters, i.e., data that belongs to the same cluster should be as similar as possible and 2) *Heterogeneity* between clusters, i.e., data that belongs to different clusters should be as different as possible. This concept has been found useful [2–4] in realizing image segmentation wherein a raw image is taken as the input data, then clustering operates on the feature space to label sufficiently similar and adjacent pixels. After the clustering process, the labeled results are mapped onto the image plane to obtain the final segmented objects.

On the other hand, in hybrid-based segmentation methods [5] or the so called split-and-merge techniques, a preprocess is initially invoked to dissect the input image into a set of small primitive regions referred to as initial image tessellation. One well known such pre-process is the morphological watershed transform [6, 7] which has drawn great attentions from researches for two main reasons: its easiness in implementation, and more importantly, it generates a completely closed contour. However, due to the keen sensitivity to local variance the technique suffers a major drawback known as over-segmentation problem [6], hence an effective merging process is needed in order to achieve valid segmentation. To this end, Vincent et al. [8] used an undirected RAG (Region Adjacency Graph) [9] to sequentially merge small regions pre-generated by watershed analysis. Two regions having the minimum edge are searched globally and merged. Such sequential operation is rather time-consuming. Haris et al. [10] proposed the nearest neighbor graph (NNG) region merging method. Unlike the work of Vincent et al. [8], not all RAG is kept in the heap; instead only a small portion of it is needed. But the improvement is still rather limited as NNG also conducts merging in a sequential fashion. Another shortcoming of sequential merging is the lack of spatial information needed for estimating the merging criterion, resulting in false contours and less accurate segmentation. Obviously, this undesired effect is further deteriorated by the more number of initial primitive regions.

To overcome the aforementioned shortcomings encountered in conventional split-and-merge techniques, and to further improve merging accuracy while resolving the time consuming problem, we propose a novel approach called self-organizing fusion neural network (SOFNN) to achieve image segmentation in a parallel fashion. Unlike *k*-means algorithm [4], SOFNN does not need to pre-specify the final number of objects. Each individual neuron is associated with a working feature to characterize its gray or color features. Another key attribute of SOFNN is the association of each target neuron with two statistical parameters α_i and β_i which are used to adjust the learning rate in a counteracting manner, namely one favors region-merging while the other favors region-splitting. After counteracting learning, the working features are simultaneously updated, and features of adjacent regions will become more similar, and tend to blend to each other, hence the name fusion. During the training process, local statistics pertaining to current adjacency information and relative similarities are collected to update the values of α_i and β_i. We will show that because of the counteracting effect between α_i and β_i, the learning can be iterated into a convergence state that represents the final segmentation result that is free of over-segmentation and under-segmentation problems. In addition, in order to further improve segmentation accuracy and the ability of noise tolerance, an extended adjacency strategy is involved in updating the connection weight between two adjacent neurons. The strategy emphasizes that not only the immediate regions of the target region but their neighboring regions are also accounted for. Therefore, a wider range of local statistics is collected to compute for a more reliable merging criterion, which not only improves noise immunity, but also help avoiding trapping in a local minimum.

Thus, the main contributions of this paper are the proposal for a novel approach that incorporates the self-organizing fusion neural network, a concept of adjacency extension, and accurate identification of regions for performing image segmentation, characterizing properties such as convergence and data-driven training parameters, and contrasting the performance with other merging methods and further validation through simulation studies. The rest of the paper is organized as follows. Section 2 elaborates the three major phases of SOFNN: *counteracting learning (i.e., simultaneous updating), fusion, and merging*, they perform segmentation in a manner of relay race. In Section 3, we demonstrate how to apply SOFNN to perform segmentation for fairly complex real images. To further exploit its applicability, SOFNN is also used as a pre-process prior to clustering gene-expression data, results show that the scale variation of gene-expression data can be reduced without sacrificing essential information such as inherent hierarchical structure of data. In Section 4, important properties of SOFNN such as convergence analysis and parametric characterizations are addressed. Finally, concluding remarks and discussions are given in Section 5.

2. Self-Organizing Fusion Neural Networks

Network architecture of the proposed SOFNN is shown in **Fig. 1**. Initially the average intensity of all pixels in each primitive region dissected by a pre-process, e.g. the watershed transform, is used as an input working feature to be associated with an individual neuron (indicated by a black circle in **Fig. 1**). The similarity between an arbitrary neuron and its adjacent neuron is used as the connection weight. After applying a counteracting learning strategy to simultaneously update the working features, adjacent regions will become more similar in feature space (this means the corresponding neurons will get closer to each other). Interestingly, the working features of adjacent regions with similar features tend to blend to each other.

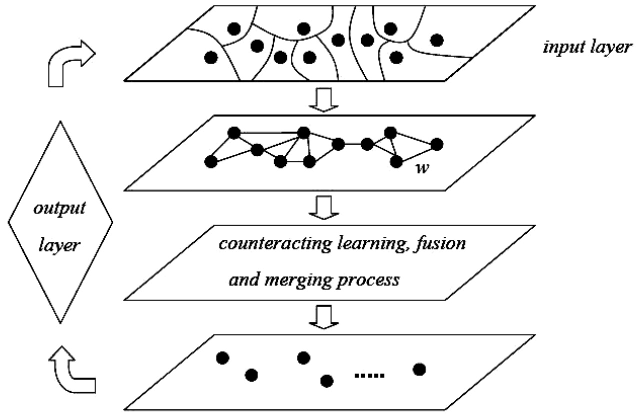

Fig. 1. Architecture of SOFNN.

This fusion phenomenon is one of many key attributes of SOFNN; it acts the underlying mechanism that facilitates the subsequent self-organizing merging. That is, if some working features are similar enough, their associated neurons will be combined into a neuron, meaning the pertaining regions will be merged. In this context, fusion desirably serves the role of identifying objects comprising a meaningful, resolvable object. Note that after the merging, RAG will be updated in order to track the new adjacency configuration for the next training iteration.

2.1. Fusion in Feature Space

Our idea of segmentation is rooted in the observation that regions comprising a resolvable object are at least near-homogeneous in the feature space, and more importantly, they must be adjacent in the image plane. Thus, it is vital to quickly search adjacent regions for each target region and accurately identify which are qualified to be merged. In addition, it is necessary to keep track of adjacency information if there are regions merged during the preceding iterations, the statistical parameters used in this work are closely related to the adjacency information. All these operations require an efficient data structure for representing adjacency, an ideal choice of data structure would be the undirected Region Adjacency Graph (RAG) [9]. The interpretation of an edge in RAG is this: the more two adjacent regions are similar, the more value the edge is. RAG can iteratively update the adjacent information accurately.

After RAG is built, an arbitrary regionR_i is characterized by a working feature $I^t(R_i)$ defined as the average intensity of all pixels in R_i at iteration t, value of $I^t(R_i)$ will be iteratively updated until convergence. Because each neuron corresponds to a region in SOFNN, the similarity between an arbitrary region R_i and its immediate neighboring regions $R_j, \{R_j, j = 1, 2, \ldots, n-1\}$ is characterized by the free parameter of connection weight prescribed as

$$w_{ij}^t = \frac{1}{e^{\lambda(t)\times(|I^t(R_i)-I^t(R_j)|)}},\ 0 < w_{ij}^t \leq 1 \quad . \ . \ . \quad (1)$$