Object scale selection of hierarchical image segmentation with deep seeds

Zaid Al-Huda\textsuperscript{1,2} \mid Bo Peng\textsuperscript{1,2} \mid Yan Yang\textsuperscript{1,2} \mid Riyadh Nazar Ali Algburi\textsuperscript{3}

\textsuperscript{1} School of Information Science and Technology, Southwest Jiaotong University, Chengdu, Sichuan, China
\textsuperscript{2} National Engineering Laboratory of Integrated Transportation Big Data Application Technology, Southwest Jiaotong University, Chengdu, Sichuan, China
\textsuperscript{3} School of Mechanical Engineering, Southwest Jiaotong University, Chengdu, Sichuan, China

Abstract
Hierarchical image segmentation is a prevalent technique in the literature for improving segmentation quality, where the segmentation result needs to be searched at different scales of the hierarchy to identify objects represented from various scales. In this paper, a novel framework for improving the quality of object segmentation is presented. To this end, the authors first select the optimal segments among several hierarchical scales of the input image using simple mid-level features and dynamic programming. Simultaneously, deep seeds are localised on the input image for the foreground and background classes using a deep classification network and a saliency network, respectively. Then, a graphical model is constructed as a set of nodes that jointly propagate information from deep seeds to unmarked regions to obtain the final object segmentation. Comprehensive experiments are performed on different datasets for popular hierarchical image segmentation algorithms. The experimental results show that the proposed framework can significantly improve the quality of object segmentation at low computational costs and without training any segmentation network.

1 \ INTRODUCTION

Image segmentation is an important technique in computer vision, which divides an image into several segments on the basis of specific properties, such as uniform colors or similar textures. Comprehensive segmentation algorithms have been proposed in the literature, however an effective division of an object into meaningful segments to visualise human perception remains an open challenge [1–5]. Different definitions of the term “meaningful” have been reported, which leads to different approaches to segment an image properly. Some researchers prefer segmenting an image into several segments, others tend to identify only a few segments of image contents [4, 6]. In the latter scenario, a single segmentation result can only be generated by an algorithm, which may be insufficient in solving the image segmentation problem.

To this end, researchers [7–11] have investigated the partitioning of an image into a single multiscale structure that aims to capture objects at all scales. As a dual representation, the Ultrametric Contour Map (UCM) was used [8–11] to explain the scales in terms of object boundaries, in which each boundary was weighted in real value to represent its scale. Although the bottom-up hierarchy structure makes it easier to capture objects at all scales, using a single scale from the hierarchies cannot guarantee that all objects are segmented in high quality [12]. The use of hierarchical segmentation increases the probability, that an entire object or part of an object is contained in a particular hierarchy scale. However, hierarchical algorithms exhibit a shortcoming in instability. Their results were based on the space selection and parameters of a feature because hierarchies are built on low-scale features (e.g. edges, colors) and the scale of an object was not imposed to achieve coherence [12]. Nonetheless, it is important to find the right scales for various objects. The method presented in [12] attempted to align scales to the same scale/level for all objects. A classifier was trained to realign object scales on the basis of a large pool of low- and mid-level features. In this paper, we propose a novel framework, as a post-processing step, to learn the concept of object scale from the hierarchy’s mid-level features with dynamic programming to search for an optimal tree cut.
In recent studies [13–15], deep image classification networks were trained just from image-level labels and were successfully employed to retrieved seeds on object localisation. These seeds can be very useful to guide graphical model to propagate information from deep seeds to unmarked regions to obtain the final object segmentation. In this paper, we use a classification network to get activation maps, then we combine the activation maps with saliency map [16] to produce deep seeds.

Motivated by the hierarchical segmentation methods and deep image classification networks, we introduce a novel framework to improve the quality of object segmentation by combining three major steps. First, the input image is passed through a hierarchical segmentation algorithm to predict the boundary map from a hierarchical segmentation tree. Then, dynamic programming is performed to search for an optimal tree cut based on the segmentation qualities. Simultaneously, the input image is passed through a deep classification network and a saliency network [16] to generate deep deeds. Lastly, a graphical model based on spatial constraints, appearance, and semantic content is proposed to spread information from seeds to unmarked regions. We name the proposed method as “Object Scale Selection with Deep Seeds (OSS-DS)” for object segmentation.

The major contributions of this work are as follows: First, a novel method is provided for processing the hierarchy of segmentation by selecting regional thresholds. This method does not rely on any ground truth, thus can be used in online processing with a small pool of simple features. Second, a graphical model is designed and the corresponding loss function is solved to propagate information from unmarked regions from deep seeds and obtain the final object segmentation. Third, comprehensive experiments on several state-of-the-art hierarchical segmentation algorithms are performed to validate the proposed method. The remaining sections of this paper are organised as follows: Section 2 provides a review of related work. Section 3 presents the proposed method. Section 4 describes the experiments performed on open datasets by using state-of-the-art algorithms for comparison. Section 5 concludes the study.

2 | RELATED WORK

2.1 | Hierarchical segmentation

One of the drawbacks of unsupervised segmentation methods is that parameters frequently need to be adjusted to control the number of segmentation regions. In practice, for all objects and several images, no single threshold or parameter fits well [17]. Instead of working on a single scale, a more flexible solution is to use the intermediate results of the merging process to build a partition hierarchy that is often described as a tree structure. Hierarchical segmentation methods [8–11] address this issue through output rather than multiple segmentations nesting in a tree, such that objects will be properly segmented at certain tree levels. These algorithms can produce a hierarchy of regions grouped into a sequence of granularity with pixels. Meanwhile, this region’s spatial relationship is preserved.

Segmentation quality was predicted using a selected classification scheme in various studies. Ren et al. [18] used a linear classifier based on Gestalt features [19] to distinguish good and bad segmentations. Their negative training data were generated through the random placement of a ground truth mask over an image. Peng et al. [20] adopted a similar idea for selecting parameters based on interactive segmentation by choosing $\lambda$ in the graph cut. They selected the highest-quality segmentation by computing segmentation using different $\lambda$ values. Instead of training a classifier to use a large pool of cues similar to that in [12], we use only three middle-scale cues to evaluate segmentation quality. As a post-processing step, this method is used to evaluate segmentation quality at different scales and is independent of the segmentation algorithm. Without imposing specific information on the segmented targets, the proposed method can be applied to any segmentation task for general purposes.

In a hierarchical structure, the segmentation results of most region-merging algorithms are not explicitly represented. The selection of segmentation and scale is performed simultaneously on the basis of certain regional cues; therefore, integrating object information globally across different scales is a challenging task. Finding the right scales for several objects is important. In the method in [12], all objects are aligned at the same scale/level. The proposed method utilises hierarchical segmentation trees to find the correct object regions. Rather than realigning object scales in [12], by searching for an optimal tree cut, we find the best regions of the object based on their segmentation qualities.

2.2 | Localisation seed with classification network

The most popular seed pixel selection method was based on a simple handcrafted criterion [21] (e.g. color, intensity, or texture). Furthermore, with regard to handcrafted features, a similarity criterion [22] is represented. Results of these settings are over-segmentation and bad segmentation. However, Kolesnikov et al. [13] proposed a method for locating seed cues to train segmentation networks in accordance with classification networks. Moreover, [13] obtained only small and sparse object seeds for supervision. To address this issue, a saliency model was proposed by Oh et al. [23] to utilise the extent of an object as additional information. Wei et al. [24] iteratively trained multiple classification networks to expand discriminatory regions as an adversarial erasing method. In our present work, we use an idea similar to that of [13] to generate seed cues for foreground classes and adopt the saliency network [16] to localise background classes. With the development of deep learning on computer vision, weakly supervised semantic segmentation methods [13–15, 25] achieves a remarkable progress. Unlike weakly supervised semantic segmentation approaches, the proposed approach does not train any segmentation network; it only uses deep classification network to set up deep seeds on the input images and adopt unsupervised process to obtain the final segmentation.
3 | PROPOSED METHOD

This section presents a technical description of the proposed method. The objective of our work is to partition an image, wherein each region describes either an object or a background, into meaningful non-overlapping regions. Figure 1 illustrates the overview of the proposed method. First, we introduce how we generated a boundary map from a hierarchical image segmentation algorithm and search for an optimal hierarchy segment based on its segmentation qualities. Then, we demonstrate how we obtained seed cues from a deep classification network and a saliency network [16]. Lastly, a graphical model is proposed to propagate the information from deep seeds to unmarked regions.

3.1 | Object segmentation and hierarchical segmentation tree

We indicate the input image as $I$. The goal is to divide image $I$ into non-overlapping regions $I = \{r_1, r_2, r_3, \ldots, r_n\}$ such that each region describes an object or a part of an object. The partitions produced by machine algorithms can be grouped into three categories based on perceptual quality. Oversegmentation contains many small pieces in the object segment, and region boundaries are typically not smooth. In contrast, undersegmentation places different objects in the same region, leading to the oversmoothness of boundaries. An appropriate segmentation is in between, wherein objects are well-segmented in exclusive regions.

Hierarchical segmentation describes an image at multiscale in which identical pixels are grouped into hierarchies of regions. Tree nodes are constructed to present meaningful data that appear in the image. In the present work, a tree structure is constructed wherein each region is represented as a tree node and the edges are represented the inclusion among regions (see Figure 2). This tree can be built using two methods: bottom-up or top-down. In the former, which is adopted in this work, to construct the hierarchy, region-merging algorithms are used. Using an initial set of regions that conforms to the finest possible partition, the algorithms typically start to construct the hierarchy. Adjacent regions are then iteratively combined, a new node
represents the output regions on the graph as parent of the merged regions. In the latter, the algorithm starts with the largest possible partition, which divides iteratively until the convergence selection criteria is achieved.

We denote a hierarchical segmentation tree as $T$, which is composed of $n$ nodes obtained from the image. A single region of the image is represented by a tree node, and the root of the tree is represented by the entire image. Furthermore, each node has at least one child (excluding the leaves) and only one parent (excluding the root). The segmentation tree also exhibits an interesting property, that is, a flat image segmentation $J_s$ can be generated by using the threshold as the tree cut. A segmentation tree and different cuts of the tree are shown in Figure 3. In the subsequent sections, we demonstrate how the quality of each region is obtained and how these qualities are used with dynamic programming to find an optimal tree cut.

### 3.2 Unsupervised segmentation quality measure

In this study, we first evaluate segmentation quality in different scales as basis for searching optimal tree cut. Studies [12, 26–28] reveal that Gestalt rules are of great significance in describing the segments. Therefore, image features which are represented by color histograms, texture histograms and the geometry size of regions are chosen. Specifically, color histograms are calculated for each channel in the CIE Lab color space. For texture histograms, the image is convolved with a bank of 38 filters [29], “which consists of an edge and a bar filters at six orientations and three scales, a Gaussian and a Laplacian of Gaussian filters”. Based on these features, the following Gestalt principles are applied to evaluate a region’s segmentation quality. Instead of computing a large pool of features for training a classifier [12], we chose to have only three features for evaluation.

#### 3.2.1 Inter-region similarity

The distinct feature distribution in adjacent regions can also indicate a good segmentation quality. In order to measure this feature, we used $\chi^2$ statistics, $H_{\text{inter}}^{\text{color}}$ and $H_{\text{inter}}^{\text{texture}}$ are the histogram bins of the region color for regions $x$ and its neighbour $y$, respectively. The inter-region color similarity of region $x$ is defined as the sum of distances from three channels to all its neighbours:

$$ f_{\text{inter}}(x) = \lambda_{\text{lab}} \sum_{m=1}^{3} \sum_{k=1}^{3} \frac{(H_{\text{lab}}^{\text{color}}(k, m) - H_{\text{lab}}^{\text{color}}(k, m))^2}{H_{\text{lab}}^{\text{color}}(k, m) + H_{\text{lab}}^{\text{color}}(k, m)}, $$

(1)

and the texture similarity is defined as:

$$ f_{\text{inter}}(x) = \lambda_{\text{tex}} \sum_{m=1}^{3} \sum_{k=1}^{3} \frac{(H_{\text{tex}}^{\text{inter}}(k, m) - H_{\text{tex}}^{\text{inter}}(k, m))^2}{H_{\text{tex}}^{\text{inter}}(k, m) + H_{\text{tex}}^{\text{inter}}(k, m)}, $$

(2)

where $H_{\text{inter}}^{\text{color}}$ and $H_{\text{inter}}^{\text{texture}}$ are the texture histogram bins for regions $x$ and $y$, respectively. And $\lambda_{\text{lab}}$ and $\lambda_{\text{tex}}$ are the coefficients. In our experiment, we set $\lambda_{\text{lab}} = 5$ and $\lambda_{\text{tex}} = 10$. A small value of $f_{\text{inter}}$ lab and $f_{\text{inter}}$ texture will represent a high quality of segmentation.

#### 3.2.2 Geometry size of the region

The large size of the object region or small number of segments may lead to a under-segmentation or an over-segmentation. In order to punish these cases, we apply the following measure for a region in the segmentation:

$$ f_{\text{geo}}(x) = \frac{N_{h_0}}{N_h} \log \left(1 + \frac{R}{R_e}\right), $$

(3)

where $N_{h_0}$ is the number of regions in the lowest-scale of hierarchy, which contains the most regions, and is used as a baseline. $N_h$ is the number of regions in the present hierarchy. $R$ is the image size, and $R_e$ is the region area. When there is an under-segmentation or an over-segmentation, the feature value would be large. For the segmentation, the quality of each region is calculated as the sum of values of all the three features. That is:

$$ Q(x) = f_{\text{inter}}(x) + f_{\text{inter}}(x) + f_{\text{geo}}(x). $$

(4)

From the above equation, we can observe that a small value of $Q(x)$ indicates a good segmentation quality for region $x$. 

**FIGURE 3** Example of (a) a segmentation tree, (b) a simple flat cut, and (c) an optimal cut
ALGORITHM 1 Dynamic programming in a tree (The forward pass)

Input: UCM map, interval \textit{Inv}.
Output: Segmentation quality at each node and its sub-tree.

Using \textit{Inv}, scales in UCM can be obtained
\[ H = \{ H_1, H_2, \ldots, H_N \} \]

1. \( i \leftarrow 1 \);
2. \( R \leftarrow H_i \), where \( R = \{ r_1, r_2, \ldots, r_k \} \);

while \( i \leq N \) do

foreach \( x \in R \) do

if \( x \) is a leaf node then

Compute \( Q(x) \);

else

Compute \( Q(x) \);

\[ Q(\tau_x) = \sum_{x_j \subseteq x} Q(x_j) \]

end

end

end

end

end

end

3.4 Deep seeds generation

We use the deep seeds generated by leveraging a deep convolution neural network, which is trained to solve the image clas-

sification task [13–15]. We use a deep classification network to set up deep seeds from discriminative regions under the supervision of the image level. However, semantic object position information cannot be provided directly from image-level labels. We utilise two methods to locate background and foreground classes (see Figure 1).

We adopt the CAMs [31] approach in the proposed method to locate the classes in the foreground. The standard VGG-19 network is used as the underlying classification network, which is initialised from the publicly available model [32]. In our experiments, we find pre-trained VGG-19 model can generate more accurate seeds compared to seeds generated using VGG-16 model. In order to make the methodology presented in [31] applicable, the architecture of VGG-19 is slightly modified. In particular, we implement the following changes in VGG-19.

To localise the background, we rely on an alternative technique from [33]. Several computer vision studies have recently concentrated on detecting the most insightful and attention-grabbing regions in a scene (i.e. salient objects). These proposed salient object detection methods [16, 34] also evolve to target on uniformly highlighting pixel-accurate saliency values. We utilise the saliency network (Basnet [16]) and normalised saliency maps to select the seeds of the background as regions with low saliency values. We adopt the normalised saliency value 0.01 as the threshold to obtain background localization cues (i.e. pixels whose saliency values are smaller than 0.01 are considered as background). Figure 1 illustrates how the deep seeds are stacked together for the foreground and background in a single-channel.

3.3 Optimal tree cut using dynamic programming

Segmentations with countless regions will not be meaningful; therefore, we discard the first few lower scales and start from the one with a medium number of regions. In existing datasets, segmentation ground truths contain less than 30 regions for an image [30]. This situation suggests that in a number of regions, a meaningful segmentation for real applications should approximately be divided into 30 regions. In our experiments, we select candidate scales from near the middle of the scales in the hierarchy to make the number of regions in \( H_1 \) (anchor) not much more than 30. To further refine segmentation quality, a smaller interval is set to select candidate scales around the anchor \( H_1 \).

For each region in the candidate scales, the quality of a region \( Q(x) \) and its sub-tree \( Q(x') \) is computed using equation (4). The optimal cut can be identified using dynamic programming in a tree structure as follows. One forward and one backward passes are included in the optimization process. From the bottom to the top of the tree, we calculate the quality at each node \( x \) and its sub-tree \( x' \) as the forward pass. The pseudo-code of Algorithm (1) is summarised in this procedure. In the backward pass, the qualities of the current node and its sub-tree are compared from the top to the bottom of the tree to determine the optimal cut. We start from the roots and move up to the leaves, level by level and node by node, to check the quality of the current node and compare it with its children. If the quality of the current node is better than its children, then we choose this node as the node of optimal path; otherwise, we find the child with the best quality (see Algorithm (2)). The complexity of this highly efficient method is \( O(n) \), where \( n \) is the total number of nodes.
3.5 The proposed graphical model

We use a graphical model to spread data to unknown regions from seeds. We build a graph on the regions of proper segment image, which we get from the optimal tree cut section. A region represents in the graph as a vertex, and the similarity between two regions represents in the graph as an edge (see Figure 1).

We denote a proper segment image as \( I \), and the set \(-r_k\) as its set of non-overlapping regions, which is satisfying \( U_k(r_k) = I \) and \( r_k \cap r_m = \emptyset \). The seeds of an input image are \( S = -s_i \) where \( s_i \) is the pixels of seeds in category \( i \) and \( 0 \leq i \leq L \) is the seed’ categories label (supposing there are \( L \) categories and \( L = 0 \) for background). For a region \( r_k \), we are looking for a category label \( 0 \leq y_k \leq L \). Therefore, we use a graph-cut optimisation framework [35] to find the final label, which minimises the following energy:

\[
E(L) = \sum_k \psi_k^{\text{seed}}(y_k | I, L) + \sum_{k,m} \psi_{k,m}(y_k, y_m | I),
\]

where \( \psi_k^{\text{seed}} \) represents a unary term including the region \( r_k \) based on deep seeds, and \( \psi_{k,m} \) represents a pairwise term between two regions \( r_k \) and \( r_m \). We define the unary term as follows:

\[
\psi_k^{\text{seed}}(y_k) = \begin{cases} 
0, & \text{if } y_k = l_i \quad \& \quad r_k \cap s_i \neq \emptyset \\
- \log \left( \frac{1}{|I|} \right), & \text{if } y_k \in \{-l_i\} \quad \& \quad r_k \cap s_i = \emptyset \\
\infty, & \text{otherwise}
\end{cases}
\]

The first condition in this equation means that when a region \( r_k \) overlaps with a seed \( s_i \), the cost is zero when this region assigned to the label \( l_i \). The second condition is to assign any seed label on this image with the same probability if a region \( r_k \) does not overlap with any seed. Where \(-l_i\) represents the number of seed labels appear on this image. To reduce false positive predictions, this exclusive information is useful.

The pairwise term \( \psi_{k,m} \) in this model represents the similarity between two regions. We adopt a pairwise term to neighbouring regions and considered a simple appearance of similarities for neighbouring regions. Then, we build the histograms of color and texture for region \( r_k \) as follows. The color histogram \( b_{\text{lab}}(r_k) \) on region \( r_k \) is built on CIE Lab color space, the color histogram is evenly divided into 30 bins. The texture histogram \( b_{\text{surf}}(r_k) \), a bank of 38 filters [29] are convolved the image including Gaussian and Laplacian of Gaussian filters, edges and bar filters with three scales and six orientations. In color/texture histograms, all bins are concatenated and normalised. This term is described as follows:

\[
\psi_{k,m}(y_k, y_m | I) = [y_k \neq y_m] \exp \left\{ - \frac{\|b_{\text{lab}}(r_k) - b_{\text{lab}}(r_m)\|^2}{\eta_{\text{lab}}^3} - \frac{\|b_{\text{surf}}(r_k) - b_{\text{surf}}(r_m)\|^2}{\eta_{\text{surf}}^2} \right\} + D_{k,m},
\]

where \([\cdot]\) is 1 if the condition is true and 0 otherwise. We set \( \eta_{\text{lab}} \) and \( \eta_{\text{surf}} \) as 10 and 15, respectively. This definition means that the neighbouring regions belonging to different labels will have higher costs if their appearance is closer, and, \( D_{k,m} \) is described as the Euclidean distance of SURF [36], and SIF [37] distance of two regions \( r_k \) and \( r_m \):

\[
D_{k,m} = \sum_{k,m} \left[ \eta_1 d_{k,m}^{\text{srf}} + \eta_2 d_{k,m}^{\text{surf}} \right]
\]

where \( \eta \) is used as a weight factor for distance adjustment and satisfies \( \eta_1 + \eta_2 = 1 \).

The optimisation of the labeling problem in equation (5) is NP-hard. It can be solved with the algorithm of expansion and swap moves [35], where the minimum cut is calculated for a defined graphical model. The nodes in the graph are defined as regions, which are connected by n-links to their neighbours.

4 EXPERIMENTS

In this section, we evaluate the proposed method on the output of hierarchical segmentation methods and analyse its effectiveness in object segmentation. The goal is to prove that the proposed method can improve object segmentation and reflect improved vision tasks at high levels.

4.1 Experiment settings

4.1.1 Dataset and evaluation

We benchmark the performance of the proposed method on the following three annotated datasets: the MSRC 21-class dataset [41]: 591 natural images with 23 categories of objects. PASCAL VOC 2012 dataset [42] contains 20 foreground object categories and 1 background category. For the segmentation task, it contains (1464) training, (1449) validation and (1456) test images. Following common practice, we use the augmentation data [43] which contains 10,582 images as training set. MS COCO dataset [44] includes 80k training images with image-level labels and 40k validation images with 81 categories. The quantitative evaluation of segmentation on these datasets is performed through using standard metrics:

**Segmentation covering (SC)** [9]: “It is a metric of the similarity between segmentations of ground truth and proposed. It gives lower value to proposed segmentations with lower similarity and gives higher value to proposed segmentations with high similarity.” It has the following definition:

\[
SC(S_{\text{prop}}, S_{\text{gt}}) = \frac{1}{N} \sum_{\|r\|} \frac{\max_{S_{\text{gt}}}|S_{\text{prop}} \cap S_{\text{gt}}|}{|S_{\text{prop}} \cup S_{\text{gt}}|},
\]

where \( N \) represents the total number of pixels in the image. \( S_{\text{gt}} \) and \( S_{\text{prop}} \) represent ground truth and proposed segmentations, respectively.
Results

4. Variation of information (VI) [9]: “It is a metric of the relative entropy between proposed segmentations and ground truth segmentations. It assigns small value to a higher similarity between proposed segmentations and ground truth segmentations”. It has the following definition:

\[ I(S_{\text{prop}}, S_{\text{gt}}) = H(S_{\text{prop}}|S_{\text{gt}}) + H(S_{\text{gt}}|S_{\text{prop}}) \]  

where \( H(S_{\text{prop}}|S_{\text{gt}}) \) and \( H(S_{\text{gt}}|S_{\text{prop}}) \) are conditional image entropies.

5. Probabilistic Rand index (PRI) [9]: “It is a metric of probability that a pair of pixels are consistently grouped between segmentations of ground truth and proposed. Probabilistic Rand Index gives higher value when segmentations of ground truth and proposed are closer.” It has the following definition:

\[ \text{PRI}(S_{\text{prop}}, S_{\text{gt}}) = \frac{1}{7} \sum_{k<m} [l_{km}p_{km} + (1 - l_{km})(1 - p_{km})], \]  

where \( l_{km} \) and \( p_{km} \) are the event that pixels \( k \) and \( m \) have the same label and its probability.

6. Jaccard similarity coefficient [45]: “The Jaccard index in the context of object segmentation is often referred to as Intersection over Union (IoU) between the machine and the ground-truth results”:

\[ \text{Jaccard Index}(P_{\text{prop}}, P_{\text{gt}}) = \frac{|P_{\text{prop}} \cap P_{\text{gt}}|}{|P_{\text{prop}} \cup P_{\text{gt}}|}, \]  

where \( P_{\text{prop}} \) and \( P_{\text{gt}} \) refer to positive pixels for proposed segmentations and ground truth, respectively. For a review of this measure, see [45]. In this experiment, we set Jaccard Index > 0.5 for better vision.

All at Optimal Dataset Scale (ODS). We also use mean intersection over union (mIoU).

4.1.2 | Segmentation techniques

Five popular hierarchical segmentation algorithms are selected to produce UCMs to validate the proposed method, we chose them due to their publicly available code and the-state-of-the-art performance:

1. gPb-UCM [9]: A commonly used method of hierarchical segmentation. This method produced different scales of segmentation by applying different thresholds to the UCM.

2. MCG [10]: A framework called the Multi-scale Combinatorial Grouping MCG that uses multi-scale information from UCM and fast-normalised cuts technique to get region segmentation.

3. SCG [10]: A single resolution version of the MCG. It has competitive results and is faster than the MCG.

4. PMI [11]: An algorithm for unsupervised boundary detection named Pointwise Mutual Information (PMI), which uses different local information embedded in a matrix of affinity to predict whether two pixels belong to the same object or not. Then segments and boundaries are obtained by applying spectral clustering.

5. COB [8]: A recent work called Convolutional Oriented Boundaries (COB) uses deep convolutional networks to predict high-quality boundary maps from images. It combines features from several layers to compute oriented contours and region hierarchies.

5 | Results

In general, hierarchical segmentation algorithms do not directly provide high-level segmentation results for object detection. However, the extraction of object regions from hierarchical segmentation results is possible through several post-processing steps. The aim of segmentation is not for a specific application, but providing well grouped pixels for further analysis. The hierarchical segmentation is originally designed for this purpose. In this section, experiments are performed to validate the improvement produced by the proposed algorithm. For comparison, the ODS is applied to choose the optimal scale of segmentation in UCM maps. Tables 1 and 2 show the evaluation results of different hierarchy methods on MSRC and Pascal VOC 2012 datasets, respectively. These tables show segmentation results with and without applying the proposed method as post-processing step for each hierarchy methods; also we compare our results to other hierarchical segmentation algorithms results, which provided their results for these datasets. The improvements achieved by the proposed method over SC and VI are obvious across different segmentation methods. For PRI, it is known to suffer from a small range [9], whereas the values of different methods are close in the test. For COCO dataset, only MCG, SCG and COB provided their results on COCO dataset; so Table 3 shows the evaluation results of these hierarchy methods. In particular, the segmentation results of
we set

\[
\text{J} \geq 0.5
\]

for better vision. Therefore, when a method has a wide range on the x-axis, that means most images have Jaccard Index values more than 0.5, and when these values are high on the y-axis, this means a method gives the best results. The proposed method based on COB outperforms the other methods used in these datasets. Overall, we demonstrate that the proposed algorithm can produce a consistent improvement on the segmentation quality. Additionally, Figure 4 shows that the segmentation quality of optimal cut of UCMs is better than the superpixel generating methods: SLIC [57], GraphSeg [40]. Also, it is better than the segmentation of using only single scale of UCMs. Afterward, optimal cut with deep seeds exhibits high-precision segmentation performance. Moreover, we compare the results by applying these post-processing steps to superpixels [40]. Figure 6 shows effectiveness of using optimal cut of high-quality region hierarchies to produce high-precision segmentation performance compared to using superpixels [40].

In the following sections, we will compare our approach to other approaches on the three datasets. Note that most weakly supervised approaches give their results only on PASCAL 2012 VOC dataset and MSRC dataset used by superpixels approaches, so we compare our approach to both categories. Also, few approaches give their results on COCO dataset. Therefore, we cannot compare the three datasets on the same approaches.

### 4.2.1 Comparison on MSRC-21 dataset

We first provide several examples of segmentation on MSRC-21 to illustrate visually the performance of the proposed method. Some of the results of the proposed method on the MSRC-21 dataset are presented in Figure 7. The proposed method produces better segmentation results based on these segmentation examples. Comparing the performance of each algorithm on MSRC-21 intuitively, the mean intersection over union (mIoU) and pixel accuracy of all the image labels are provided in Table 4 for the proposed method (OSS-DS based on COB) and other algorithms for comparison. Table 4 indicates that the proposed method outperforms the other methods. Where, each column represents different algorithm accuracies for each semantic class, and the average accuracy of all the classes is provided in the last column of the table. The best segmentation performance in the tables is written in bold font.

### 4.2.2 Comparison on PASCAL 2012 VOC dataset

We first present several segmentation examples on the VOC 2012 dataset to visually demonstrate the performance of the proposed method. Some of the results of the proposed method on the VOC 2012 dataset are provided in Figure 8. The segmentation examples in this figure indicate that the proposed method achieves better segmentation results. We also compare the performance of the proposed method with weakly supervised semantic segmentation methods on VOC 2012. Unlike these methods, the proposed method does not train any segmentation network. These approaches take more computational time and require special equipment as a high-quality GPU. Table 5 provides the (mIoU) and pixel accuracy of all the image labels for the proposed method (OSS-DS based on COB) and the other compared methods. According to Table 5, the proposed method achieves comparable results to weakly supervised semantic segmentation methods.

### 4.2.3 Comparison on COCO dataset

Next, we perform a set of experiments on COCO dataset to demonstrate the generality of the proposed method. In con-
contrast with PASCAL VOC, most of the COCO samples are collected from non-iconic images in a complex natural context. We provide the per-class IoU of BFBP [50], SEC [13], DSRG [14], and the proposed method OSS-DS in Table 6. The proposed method achieves comparable results in large-size categories, such as outdoor, animal, and vehicle. However, in small-size categories, such as indoor and kitchenware, this method performs poorly because we cannot place deep seeds precisely in small-size categories, such that seeds from large-size categories spread their information to small-size categories.
**FIGURE 6** Effectiveness of using the optimal cut with deep seeds. From left to right: original image, ground truth, optimal cut, optimal cut with deep seeds, superpixel [40] and superpixel with deep seeds.

**FIGURE 7** A comparison of object segmentation results on MSRC-21. From left to right: original image, ground truth, UCM map, optimal cut and OSS-DS. From top to bottom: gPb-UCM, MCG, SCG, PMI and COB.
### Comparison of mIoU scores (%) of different methods on MSRC-21 dataset

|       | house | grass | tree | cow | sky | sheep | plane | water | face | bike | flower | sign | bird | book | chair | road | cat | dog | body | boat | mIoU |
|-------|-------|-------|------|-----|-----|-------|-------|-------|------|------|--------|------|------|------|-------|------|-----|-----|------|------|------|
| SR [47] | 53    | 63    | 68   | 74  | 67  | 64    | 65    | 52    | 64   | 60   | 47     | 53   | 60   | 58   | 67    | 57   | 47  | 62  | 57   | 13   | 58   |
| STC [48] | 60    | 65    | 73   | 65  | 57  | 54    | 80    | 47    | 65   | 73   | 53     | 68   | 64   | 60   | 65    | 62   | 66  | 56  | 67   | 59   | 11   |
| SCR [49] | 53    | 80    | 65   | 81  | 76  | 76    | 70    | 59    | 68   | 75   | 48     | 60   | 70   | 56   | 65    | 49   | 58  | 60  | 48   | 59   | 14   |
| AIS-SIL [39] | 74    | 95    | 89   | 87  | 96  | 83    | 73    | 93    | 86   | 75   | 88     | 96   | 82   | 56   | 93    | 89   | 59  | 72  | 78   | 56   | 82   |
| OSS-DS | 83    | 91    | 92   | 95  | 88  | 91    | 85    | 86    | 88   | 85   | 75     | 87   | 85   | 90   | 84    | 78   | 82  | 85  | 89   | 82   | 87   |

**FIGURE 8** A comparison of object segmentation results on Pascal VOC2012 validation set. From left to right: original image, ground truth, UCM map, optimal cut and OSS-DS. From top to bottom: gPb-UCM, MCG, SCG, PMI and COB.

#### 4.2.4 Comparison on lung dataset

Lastly, we perform a set of experiments on lung dataset to verify the performance of the proposed method in medical field. Segmentation is an essential step during the diagnostic, and treatment stages. Having an accurate segmentation algorithm can make a big difference in patients’ life [58, 59]. Lung CT image segmentation is a necessary initial step for lung image analysis, it is a prerequisite step to provide an accurate lung CT image analysis such as lung cancer detection. Early detection of lung cancer could reduce the mortality rate and increase the patient’s survival rate when the treatment is more likely curative. However, designing an effective lung segmentation method is a challenging problem, especially for abnormal lung parenchyma tissue, where the nodules and blood vessels need to be segmented with the lung parenchyma. Moreover, the lung parenchyma needs to be separated from the bronchus regions that are often confused with the lung tissue.

A lung segmentation dataset is a collection of 2D and 3D images with manually segmented lungs at the Kaggle Data [60]. We use 70% of the data as the train set and the remaining 30% as the test set. The size of each image is $512 \times 512$. For this dataset, we modified the U-Net [61] to perform image classification by replacing the last $1 \times 1$ convolution with a global average pooling layer followed by a fully connected layer that outputs a single number $\in (0, 1)$. The global average pooling layer
es the mean value across all spatial dimensions of the input and is used to recover a class activation map. To get deep seeds from a network trained for image classification, we used class activation maps (CAMs) following the work of Zhou et al. [62]. Then, a hard threshold is applied to the heat map to obtain the discriminative object regions. We threshold the corresponding heat map to produce localisation cues for each foreground class by 20% of its maximum value. Lastly, the deep seeds guided a graphical model constructed over the optimal segments to propagate information from deep seeds to unmarked regions to obtain the final object segmentation. Figure 9 shows some segmentation outputs of the proposed method for lung dataset. For comparison, the ODS is applied to choose the optimal scale of segmentation in UCM maps. Table 7 shows the evaluation results of different hierarchy methods on lung dataset. It shows our results based on COB outperform the other algorithms,
TABLE 6 Comparison of mIoU scores (%) of different methods on COCO validation set

| Cat. | Class          | SEC  | BFBP | DSRG | OSS-DS |
|------|----------------|------|------|------|--------|
| BG   | Background     | 74.3 | 68.8 | 80.6 | 85.5   |
| P    | Person         | 43.6 | 27.5 | 61.5 |        |
| Vehicle | Bicycle    | 24.2 | 18.2 | 30.4 | 24.5   |
|      | Car            | 15.9 | 7.2  | 22.1 |        |
|      | Motorcycle     | 52.1 | 40.5 | 54.2 | 49.3   |
|      | Airplane       | 36.6 | 32.0 | 45.2 | 51.8   |
|      | Bus            | 37.7 | 39.2 | 38.7 | 46.2   |
|      | Train          | 30.1 | 26.5 | 33.2 | 38.6   |
|      | Truck          | 24.1 | 17.5 | 25.9 | 28.9   |
|      | Boat           | 17.3 | 16.5 | 20.6 | 32.7   |
| Outdoor | Traffic light | 16.7 | 3.9  | 16.2 | 14.8   |
|      | Fire hydrant   | 55.9 | 33.1 | 60.4 | 53.2   |
|      | Stop sign      | 48.4 | 28.4 | 51.0 | 48.3   |
|      | Parking meter  | 25.2 | 25.5 | 26.3 | 23.0   |
|      | Bench          | 16.4 | 12.4 | 22.3 | 20.2   |
| Animal | Bird          | 34.7 | 31.1 | 41.5 | 49.8   |
|      | Cat            | 57.2 | 52.8 | 62.2 | 68.9   |
|      | Dog            | 45.2 | 44.1 | 55.6 | 63.4   |
|      | Horse          | 34.4 | 34.2 | 42.3 | 54.7   |
|      | Sheep          | 40.3 | 38.0 | 47.1 | 53.1   |
|      | Cow            | 41.4 | 42.1 | 49.3 | 54.3   |
|      | Elephant       | 62.9 | 65.2 | 67.1 | 69.5   |
|      | Bear           | 59.1 | 57.0 | 62.6 | 61.3   |
|      | Zebra          | 59.8 | 65.0 | 63.2 | 62.6   |
|      | Giraffe        | 48.8 | 55.6 | 54.3 | 47.5   |
| Accessory | Backpack  | 0.3  | 3.2  | 0.2  | 0.1    |
|      | Umbrella       | 26.0 | 28.1 | 35.3 | 17.6   |
|      | Handbag        | 0.5  | 1.1  | 0.7  | 0.3    |
|      | Tie            | 6.5  | 5.5  | 7.0  | 0.5    |
|      | Suitcase       | 16.7 | 21.3 | 23.4 | 20.6   |
| Sports | Frisbee       | 12.3 | 5.6  | 13.0 | 12.2   |
|      | Skis           | 1.6  | 1.0  | 1.5  | 1.6    |
|      | Snowboard      | 5.3  | 2.8  | 16.3 | 7.8    |
|      | Sports ball    | 7.9  | 1.9  | 9.8  | 8.5    |
|      | Kite           | 9.1  | 10.3 | 17.4 | 11.9   |
|      | Baseball bat   | 1.0  | 1.7  | 4.8  | 2.5    |
|      | Baseball glove | 0.6  | 0.5  | 1.2  | 0.7    |
|      | Skateboard     | 7.1  | 6.6  | 14.4 | 12.3   |
|      | Surfboard      | 7.7  | 3.3  | 13.5 | 11.5   |
|      | Tennis racket  | 9.1  | 5.5  | 6.8  | 8.9    |
|      | Bottle         | 13.2 | 9.6  | 22.3 | 14.3   |
| Kitchen | Wine glass   | 22.3 | 17.5 | 24.0 | 17.2   |
|      | Cup            | 17.9 | 5.6  | 20.4 | 15.2   |
|      | Fork           | 1.8  | 0.5  | 0.0  | 0.0    |

and PMI receives the largest improvement. Also, Table 8 shows the quantitative results of using the proposed method as post-processing compared with only use U-Net to obtain final results. It is clear that the proposed method gives more accurate segmentation, which can help in cancer detection and treatment.

At the end of these experiments, we found that the quality of segmentation generated by hierarchical segmentation algo-
5 CONCLUSION

In this work, we addressed the problem of object segmentation by using the output of hierarchical image segmentation algorithms with deep seeds as a post-processing method. To achieve high-quality segmentation, simple mid-level features were used to describe the quality of an object region. In addition, an optimal segment from several hierarchical scales of the input image was selected using dynamic programming. Then, deep seeds were located in the input image using a deep classification network and saliency network. Lastly, graph labelling was resolved via expansion and swap move algorithms to propagate information from seeds to unmarked regions. We performed experiments on four segmentation datasets. The results showed that the proposed method achieves higher-precision image segmentation results compared with state-of-the-art algorithms. For our future work, we will explore more features to describe the quality of regions. Furthermore, we will focus on developing more effective strategies to improve the quality of seeds, particularly for small-size categories.

ACKNOWLEDGEMENTS

This work was supported by the National Science Foundation of China (Nos. 61772435 and 61976247).

## REFERENCES

1. Wu, Z., et al.: Hierarchical segmentation evaluation of region-based image hierarchy. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing 12(8), 2718–2727 (2019)
2. Jiang, Q., et al.: Weakly-supervised image semantic segmentation based on superpixel region merging. Big Data and Cognitive Computing, 3(2), 31 (2019)
3. Qu, S., et al.: Supervised image segmentation based on superpixel and improved normalised cuts. IET Image Processing 13(12), 2204–2212 (2017)
4. Syu, J.H., et al.: Hierarchical image segmentation based on iterative contraction and merging. IEEE Transactions on Image Processing. 26(5), 2246–2260 (2017)
5. Alghodhaifi, H., Lakshmanan, S: Simulation-based model for surrogate safety measures analysis in automated vehicle-pedestrian conflict on an urban environment. In: Autonomous Systems: Sensors, Processing, and Security for Vehicles and Infrastructure 2020, vol. 11415, pp. 1141504. SPIE, Bellingham, WA (2020)
6. Cho, S.I., et al.: Human perception-based image segmentation using optimising of colour quantisation. IET Image Processing. 8(12), 761–770 (2014)
7. Liu, T., et al.: Image segmentation using hierarchical merge tree. IEEE Transactions on Image Processing. 25(10), 4596–4607 (2016)
8. Maninis, K.K., et al.: Convolutional oriented boundaries: From image segmentation to high-level tasks. IEEE Transactions on Pattern Analysis and Machine Intelligence. 40(4), 819–833 (2017)
9. Arbelaez, P., et al.: Contour detection and hierarchical image segmentation. IEEE Transactions on Pattern Analysis and Machine Intelligence. 33(5), 898–916 (2010)
10. Pont, Tuset, J., et al.: Multiscale combinatorial grouping for image segmentation and object proposal generation. IEEE Transactions on Pattern Analysis and Machine Intelligence. 39(1), 128–140 (2016)
11. Isola, P., et al.: Crisp boundary detection using pointwise mutual information. In: Fleet D., Papilla T., Schiele B., Tuytelaars T. (eds.) Computer Vision–ECCV 2014, pp. 799–814. Springer International, Cham (2014)
12. Shen, Y., et al.: Scale-aware alignment of hierarchical image segmentation. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, pp. 364–372. Piscataway, NJ (2016)
13. Kolesnikov, A., Lampert, C.H.: Seed, expand and constrain: Three principles for weakly-supervised image segmentation. In: Leibe B., Matas J., Sebe N., Welling M. (eds.) Computer Vision–ECCV 2016, pp. 695–711. Springer International, Cham (2016)
14. Huang, Z., et al.: Weakly-supervised semantic segmentation network with deep seeded region growing. 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, pp. 7014–7023. (2018)
15. Wang, X., et al.: Weakly-supervised semantic segmentation by iteratively mining common object features. 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, pp. 1354–1362. (2018)
16. Qin, X., et al.: Basnet: boundary-aware salient object detection. 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, pp. 7471–7481. (2019)
17. Yuan, C., et al.: Image segmentation based on modified superpixel segmentation and spectral clustering. The Journal of Engineering. 2018(16), 1704–1711 (2018)
18. Ren, X., Malik, J.: Learning a classification model for segmentation. Proceedings ninth IEEE International Conference on Computer Vision, Nice, France, 1, 10–17 (2003)
19. Palmer, S.E.: Vision Science: Photons to Phenomenology. MIT Press, Cambridge, MA (1999)
20. Peng, B., Veksler, O.: Parameter selection for graph cut based image segmentation. In: Everingham M., Needham C., (ed.) Proceedings of the british machine vision conference pp. 16.1–16.10. BMVA Press (2008)

## ORCID

Zaid Al-Huda 🌐 https://orcid.org/0000-0002-8920-7635
21. Shih, F.Y., Cheng, S.: Automatic seeded region growing for color image segmentation. Image and Vision Computing 23(10), 877–886 (2005)
22. Borges, V.R., et al.: Region growing for segmenting green microalgal images. IEEE/ACM Transactions on Computational Biology and Bioinformatics (TCBB), 15(1), 257–270 (2018)
23. Oh, S.I., et al.: Exploiting saliency for object segmentation from image level labels. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Honolulu, HI, pp. 5038–5047. (2017)
24. Wei, Y., et al.: Object region mining with adversarial erasing: a simple classification to semantic segmentation approach. 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Honolulu, HI, pp. 6488–6496. (2017)
25. Wang, Q., et al.: Weakly supervised adversarial domain adaptation for semantic segmentation in urban scenes. IEEE Transactions on Image Processing, 28(9), 4376–4386 (2019)
26. Cai, Q., et al.: An adaptive-scale active contour model for inhomogeneous image segmentation and bias field estimation. Pattern Recognition, 82, 79–95 (2018)
27. Li, K., et al.: Iterative image segmentation with feature driven heuristic four-color labeling. Pattern Recognition, 76, 69–79 (2018)
28. Li, Y., et al.: Scale selection for supervised image segmentation. Image and Vision Computing. 30(12), 991–1003 (2012)
29. Leung, T., Malik, J.: Representing and recognizing the visual appearance of materials using three-dimensional textures. International Journal of Computer Vision 43(1), 29–44 (2001)
30. Martin, D.R., et al.: An empirical approach to grouping and segmentation. Computer Science Division, University of California Berkeley (2003)
31. Zhou, B., et al.: Learning deep features for discriminative localization. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, pp. 2921–2929. (2016)
32. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. International Conference on Learning Representations (2015)
33. Jiang, H., et al.: Salient object detection: A discriminative regional feature integration approach. 2013 IEEE Conference on Computer Vision and Pattern Recognition, Portland, OR, pp. 2083–2090. (2013)
34. Wang, Q., et al.: Saliency detection by multiple-instance learning. IEEE Transactions on Cybernetics. 43(2), 660–672 (2013)
35. Boykov, Y., et al.: Fast approximate energy minimization via graph cuts. IEEE Transactions on Pattern Analysis and Machine Intelligence. 23(11), 1222–1239 (2001)
36. Bay, H., et al.: Surf: speeded up robust features. Computer Vision – ECCV 2006, vol. 3951, pp. 404–417. Springer, Berlin, Heidelberg (2006)
37. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. International Journal of Computer Vision. 60(2), 91–110 (2004)
38. Gao, L., et al.: Graph-without-cut: An ideal graph learning for image segmentation. Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, pp. 1188–1194. Phoenix, Arizona February 2016
39. Xie, X., et al.: Automatic image segmentation with superpixels and image-level labels. IEEE Access. 7, 10999–11009 (2019)
40. Felzenszwalb, P.F., Huttenlocher, D.P.: Efficient graph-based image segmentation. International Journal of Computer Vision. 59(2), 167–181 (2004)
41. Shotton, J., et al.: Textonboost: joint appearance, shape and context modeling for multi-class object recognition and segmentation. In: Leonards A., Bischof H., Pinz A. (eds.) Computer Vision–ECCV 2006. Lecture Notes in Computer Science, vol. 3951, pp. 1–15. Springer, Berlin, Heidelberg (2006)
42. Everingham, M., et al.: The pascal visual object classes (voc) challenge. International Journal of Computer Vision. 88(2), 303–338 (2010)
43. Harisharan, B., et al.: Semantic contours from inverse detectors. In: 2011 International Conference on Computer Vision, pp. 991–998. IEEE Computer Society, Washington, DC (2011)
44. Lin, T.Y., et al.: Microsoft coco: common objects in context. In: Fleet D., Pajdla T., Schiele B., Tuytelaars T. (eds.) Computer Vision–ECCV 2014. pp. 740–755. Springer International Publishing, Cham (2014)
45. Pont-Tuset, J., Marques, F.: Supervised evaluation of image segmentation and object proposal techniques. IEEE Transactions on Pattern Analysis and Machine Intelligence. 38(7), 1465–1478 (2015)
46. Ren, Z., Shakhnarovich, G.: Image segmentation by cascaded region agglomeration. 2013 IEEE Conference on Computer Vision and Pattern Recognition, Portland, OR, pp. 2011–2018. (2013)
47. Zhang, K., et al.: Sparse reconstruction for weakly supervised semantic segmentation. In: Rossi, F. (ed.) Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence, 1889–1895. AAAI Press, Beijing, China August (2013)
48. Wei, Y., et al.: Stc: A simple to complex framework for weakly-supervised semantic segmentation. IEEE Transactions on Pattern Analysis and Machine Intelligence. 39(11), 2314–2320 (2016)
49. Xu, X., et al.: Weakly supervised deep semantic segmentation using cnn and elm with semantic candidate regions. Complexity. 2019, 9180391 (2019)
50. Saleh, F., et al.: Built-in foreground/background prior for weakly-supervised semantic segmentation. In: Leibe B., Matas J., Sebe N., Welling M. (eds.) Computer Vision–ECCV 2016. pp. 413–432. Springer International Publishing, Cham (2016)
51. Hong, S., et al.: Decoupled deep neural network for semi-supervised semantic segmentation. Advances in Neural Information Processing Systems, pp. 1495–1503. NIPS 2015
52. Zeng, Y., et al.: Joint learning of saliency detection and weakly supervised semantic segmentation. 2019 IEEE/CVF International Conference on Computer Vision (ICCV), Seoul, Korea (South), pp. 7222–7232. (2019)
53. Qi, X., et al.: Augmented feedback in semantic segmentation under image level supervision. In: Leibe B., Matas J., Sebe N., Welling M. (eds.) Computer Vision – ECCV 2016, pp. 90–105. Springer International Publishing, Cham (2016)
54. Chaudhry, A., et al.: Discovering class-specific pixels for weakly-supervised semantic segmentation. In: Tae-Kyun K., Stefanos Z., Gabriel B., Krystian M. (eds.) Proceedings of the British Machine Vision Conference (BMVC) pp. 20.1–20.13. BMVA Press (2017)
55. Roy, A., Todorovic, S.: Combining bottom-up, top-down, and smoothness cues for weakly supervised image segmentation. 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, pp. 7282–7291. (2017)
56. Kim, D., et al.: Two-phase learning for weakly supervised object localization. 2017 IEEE International Conference on Computer Vision (ICCV), Venice, pp. 3554–3563. (2017)
57. Achanta, R., et al.: Slic superpixels compared to state-of-the-art superpixel methods. IEEE Transactions on Pattern Analysis and Machine Intelligence. 34(11), 2274–2282 (2012)
58. Alghodhaifi, H., et al.: Predicting invasive ductal carcinoma in breast histology images using convolutional neural network. 2019 IEEE National Aerospace and Electronics Conference (NAECON), Dayton, OH, USA, pp. 374–378. (2019)
59. Alghodhaifi, H.: Modeling of the Liver Deformation During Breathing in Radiation Therapy. Master thesis, University of Michigan (2017)
60. Mader, K.S.: Finding and Measuring Lungs in CT Data, Version 2. Retrieved July 10, 2020 from https://www.kaggle.com/kmader/finding-lungs-in-ct-data (2017)
61. Ronneberger, O., et al.: U-net: Convolutional networks for biomedical image segmentation. In: Navab N., Hornegger J., Wells W., Frangi A. (eds.) Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015. pp. 234–241. Springer International Publishing, Cham (2015)
62. Zhou, B., et al.: Learning deep features for scene recognition using places database. Advances in Neural Information Processing Systems, pp. 487–495. (2014)