AW*k*S: adaptive, weighted *k*-means-based superpixels for improved saliency detection

Ashish Kumar Gupta[1] · Ayan Seal[1,2] · Pritee Khanna[1] · Ondrej Krejcar[2,3] · Anis Yazidi[4]

Abstract

Clustering inspired superpixel algorithms perform a restricted partitioning of an image, where each visually coherent region containing perceptually similar pixels serves as a primitive in subsequent processing stages. Simple linear iterative clustering (SLIC) has emerged as a standard superpixel generation tool due to its exceptional performance in terms of segmentation accuracy and speed. However, SLIC applies a manually adjusted distance measure for dis-similarity computation which directly affects the quality of superpixels. In this work, self-adjustable distance measures are adapted from the weighted *k*-means clustering (W-*k*-means) for generating superpixel segmentation. In the proposed distance measures, an adaptive weight associated with each variable reflects its relevance in the clustering process. Intuitively, the variable weights correspond to the normalization terms in SLIC that affect the trade-off between superpixels boundary adherence and compactness. Weights that influence consistency in superpixel generation are automatically updated. The variable weights update is accomplished during optimization with a closed-form solution based on the current image partition. The proposed adaptive, W-*k*-means-based superpixels (AW*k*S) experimented on three benchmarks under different distance measure outperform the conventional SLIC algorithm with respect to various boundary adherence metrics. Finally, the effectiveness of the AW*k*S over SLIC is demonstrated for saliency detection.

Keywords Superpixels · Segmentation · W-*k*-means · Distance measure

1 Introduction

In recent years, superpixel segmentation has became an integral preprocessing tool in various image processing and computer vision applications such as object detection [1, 2], recognition [3], semantic segmentation [4], image classification [5], object proposal detection [6, 7], visual tracking [8, 9], indoor seen understanding [10], and salient object detection [11–16]. Superpixel segmentation of an image partitions it into non-overlapping regions where each constituent region is a grouping of pixels that are similar in color or other low-level cues. Superpixels are found to be more natural image entities than individual pixels [5]. A superpixel segmentation algorithm is expected to address the diverse requirements comprising of boundary adherence, compactness, connectivity, and computational efficiency to be useful as a preprocessing step [1, 17–19]. These algorithms can be broadly categorized as density-based [20], graph-based [21], contour evolution [22], and cluster-based superpixels [1, 23, 24]. While most approaches have been utilized in only some specific applications, simple linear

✉ Ayan Seal
ayan@iiitdmj.ac.in

Ashish Kumar Gupta
ashishkumargupta@iiitdmj.ac.in

Pritee Khanna
pkhanna@iiitdmj.ac.in

Ondrej Krejcar
ondrej.krejcar@uhk.cz

Anis Yazidi
Anis.Yazidi@oslomet.no

[1] Department of Computer Science and Engineering, PDPM Indian Institute of Information Technology, Design and Manufacturing, Jabalpur 482005, India

[2] Center for Basic and Applied Research, Faculty of Informatics and Management, University of Hradec Kralove, Hradecka 1249, 50003 Hradec Kralove, Czech Republic

[3] Malaysia Japan International Institute of Technology, Universiti Teknologi Malaysia, Jalan Sultan Yahya Petra, 54100 Kuala Lumpur, Malaysia

[4] Artificial Intelligence Lab, Oslo Metropolitan University, 460167 Oslo, Norway

iterative clustering (SLIC) [1] has been widely applied as a preprocessor to many important computer vision problems.

The popularity of SLIC can be attributed to its characteristics of high adherence to object boundaries, explicit control over the compactness and size of superpixels, and efficiency in terms of time and space. Despite of these merits, SLIC uses a manually adjusted distance measure to encourage consistent clustering of different size superpixels [1]. This distance measure, being manually tuned, appears to affect the segmentation accuracy of generated superpixels. The selection of space-specific yet fixed normalization terms in such a distance measure is difficult given the complexity of natural images and trade-off between the conflicting requirements of boundary adherent but compact superpixels. A very few methods [1, 25, 26] have been proposed for automatic normalization of involved spaces in SLIC distance measure. Moreover, most methods attempt SLIC improvement by focusing only on the compactness parameter, a term that normalizes the color similarity between two CIE-*Lab* vectors under constrained search space.

In contrast to these methods, the proposed work applies a systematic approach for assigning and updating weights to variables in a distance measure. Motivated by the optimization function of W-*k*-means clustering [27] algorithm, distance functions proposed in this work have an adaptive term for each variable in their formulation. Due to this, the proposed distance measures can mitigate the effect of equal variable weighting problem in *k*-means algorithm. These adaptable weights provide necessary guidance about the variables for exploring desired cluster structures in data. Automatic weights update based on the current image partition replaces the previous weight values in each optimization step. Consequently, the current weights used in an adaptive, weighted distance measure reflect the relative relevances of variables and thus assist in generating boundary adherent yet compact superpixels. The main contributions of this paper are summarized below.

- Use of the proposed W-*k*-means inspired distance measures allows AW*k*S algorithm to extend SLIC to accommodate more feature spaces. Even for the original joint *Labxy* color and coordinate features, problem using two feature spaces can be extended and evaluated at individual feature components level.
- The AW*k*S update procedure for the variable weights automatically computes the terms necessary to normalize various space distances before combining. The update procedure avoids variable (feature) elimination and promotes compactness in superpixels.
- The AW*k*S algorithms under four distance measures are extensively evaluated on three challenging datasets for various qualitative and quantitative performance metrics. Further, the utility of superpixels generated by AW*k*S algorithm is demonstrated in saliency detection [11, 14, 28].

The rest of this paper is organized as follows. Section 2 reviews the background for superpixel algorithms. The proposed AW*k*S algorithm is discussed in Sect. 3. Qualitative and quantitative performance evaluation of different distance measures under AW*k*S is presented in Sect. 4. Application of the AW*k*S algorithm for saliency detection constitutes Sect. 5. Finally, Sect. 6 concludes the paper.

2 Background

2.1 Simple linear iterative clustering

Let $Y_i = [y_{i,1}, y_{i,2}, \dots, y_{i,v}]^T$ represents a feature vector consisting of v variables (attributes) for a pixel i in an input image I of size N. Then, the k-means-type clustering algorithms attempt partitioning the set $\mathbf{Y} = \{Y_1, Y_2, \dots, Y_N\}$ of N such entities into k clusters that minimize the following objective function:

$$X(\mathbf{S}, \mathbf{C}) = \sum_{l=1}^{k} \sum_{m=1}^{N} \sum_{n=1}^{v} s_l d(y_{m,n}, c_{l,n}) \tag{1}$$

where s_l is an indicator function with $s_l = 1$ if $Y_m \in S_l$ and 0 otherwise. The sets $\mathbf{S} = \{S_1, S_2, \dots, S_k\}$ and $\mathbf{C} = \{C_1, C_2, \dots, C_k\}$ represent k current data partitions and their respective cluster centers. The element-wise distance between an image pixel m and its cluster center C_l is defined as $d = (y_{m,n} - c_{l,n})^2$.

The conventional SLIC algorithm [1] solves superpixel generation as a clustering problem. The space being considered is the joint CIE-*Lab* color-image coordinate space with $Y_i = [L_i, a_i, b_i, x_i, y_i]^T$ for an image pixel i. The post-processed final clusters are then accepted as the set of desired superpixels. To guarantee identical superpixels in every run for an image I, the set $\mathbf{C}$ is initialized by sampling pixels uniformly in I with fixed grid interval $\sqrt{N/k}$. The choice of k-means clustering in SLIC [1] promotes a distance measure based on (1). A combined 5-D Euclidean distance is found to compromise with boundary accuracy and compactness of superpixels. This is caused by the very nature of conventional k-means algorithm to treat all variables equally in clustering process [27]. It, therefore, ignores normalization while combining different scale distances using (1). To mitigate this effect, the dis-similarity between a point $p = [L_p, a_p, b_p, x_p, y_p]^T$ and the qth cluster center $C_q = [L_q, a_q, b_q, x_q, y_q]^T$ is computed in [1] using a manually adjustable distance measure as follows :

$$D(p,q) = \sqrt{\left(\frac{d_c}{\eta}\right)^2 + \left(\frac{d_s}{\tau}\right)^2}, \tag{2}$$

where η (compactness parameters) and τ (set to the grid interval $\sqrt{N/k}$) normalize the color similarity (d_c) and the spatial proximity (d_s) which are computed using (3) and (4), respectively.

$$d_c = \left\| \left[L_p, a_p, b_p\right]^T - \left[L_q, a_q, b_q\right]^T \right\|_2 \tag{3}$$

$$d_s = \left\| \left[x_p, y_p\right]^T - \left[x_q, y_q\right]^T \right\|_2 \tag{4}$$

2.2 SLIC variants

The distance measure D in (2) has a space-specific normalization term for each involved space that is defined once by a user and remains fixed during the k-means optimization. For natural images, wise selection of such normalization terms is difficult due to the vague relation between η (treated as compactness parameter) and its impact on conflicting requirements of boundary adherence and compactness. To this end, Achanta et al. [1] also proposed an adaptive version of SLIC (ASLIC) that can adjust color and space proximities for each cluster based on the current image partition. ASLIC superpixels were observed to be visually more compact but slightly degraded in boundary adherence. The recently proposed variance adaptive SLIC (VASLIC) [25] utilizes a local 3×3 variance at each image pixel to generate visually boundary adherent superpixels. VASLIC severely underperforms SLIC on other evaluation metrics such as undersegmentation error and explained variance because it applies a local heuristic at each image pixel without addressing the smoothing issues. Similar to [25], the method proposed in [26] uses the non-stationarity measure (NSM) map of the input image to decide upon the superpixels compactness at pixel level. Thus, these SLIC variants estimate only the color normalization term based upon some locally varying image property and hence eliminate the need for a user to specify the compactness parameter. However, the normalization term associated with the image plane space always remains fixed in SLIC and its variants [25, 26]. Moreover, the factors effecting superpixel accuracy and compactness such as variance are themselves fixed at pixel level and do not experience any change during k-means optimization. This effectively implies that algorithms such as [25, 26] fix the color normalization term at pixel level for distance computation. In contrast to these methods, the proposed AWkS algorithm uses adaptive terms to appropriately normalize different feature spaces before combining them into a useful distance measure.

3 Adaptive, W-k-means-based superpixel segmentation

In this section, the various distance measures inspired from W-k-means objective functions are presented before discussing the AWkS segmentation algorithm.

3.1 Distance measures for AWkS

A multi-dimensional clustering problem, such as SLIC, requires a distance measure in order to detect the amount of dis-similarity between the two data points. Unlike SLIC, this work adopts a systematic approach for the incorporation of weights for variables in a distance measure along with a suitable procedure for their iterative updates during optimization. In this regard, the distance functions motivated by W-k-means clustering [27] and its variants [29] appear to be beneficial. Such algorithms have played a pivotal role in recovering the cluster structure of multi-dimensional data. The possibility of improving the accuracy of SLIC [1] superpixels by exploring upon the degree of feature relevances during objective function minimization has been attempted in this work.

Given $W = \{w_1, w_2, \dots, w_v\}$ as the weights for v variables, the W-k-means algorithm [27] minimizes the following objective function:

$$X(\mathbf{S}, \mathbf{C}, W) = \sum_{l=1}^{k} \sum_{m=1}^{N} \sum_{n=1}^{v} s_l w_n^{\beta} d(y_{m,n}, c_{l,n}) \tag{5}$$

subject to: $\sum_{n=1}^{v} w_n = 1, 0 \leq w_n \leq 1$, where the exponential β is a user-defined parameter.

In the AWkS algorithm, a distance measure is adopted from (5) and takes the following general form:

$$D(m,l) = \sum_{A \in V} w_A^{\beta} d(y_{m,A}, c_{l,A}) \tag{6}$$

where V contains disjoint subsets from the feature set of dimension v.

The advantages of using (6) are twofold. First, the term w_A^{β} serves as an adaptive normalization term for each variable subset $A \in V$. Further, all weights associated with the variable subsets in V are open for automatic updates. Second, the variable weights are computed using a closed-form solution based on every current segmentation of the input image, thus allowing the incorporation of critical cluster information in the subsequent pixel assignment step. Unlike other SLIC variants, variables relevance are computed with respect to all contributing variables and hence have an intrinsic dependency on one another.

For AWkS, it is possible to define the distance measure in two ways—at “image-level” or at “cluster-level.” For the image-level variant, the weight update procedure (as discussed in Sect. 3.2) for (6) requires the computation of the intra-cluster variable-specific distances from all image pixels to their assigned cluster centers. On the other hand, the "cluster-level" formulation is the extension of "image-level" to associate a weight for each feature in each cluster. In "cluster-level" formulation, the important variables in each cluster identify the subspace in which the cluster is discovered. Further, the subsets of important variables vary from one cluster to another resulting in subspace clustering. This requires modifying the objective function in (5) to assign a weight to each feature subset $A \in V$ at the cluster level. With $\mathbf{W}$ representing a matrix of weights having $|V| \times k$ dimensionality, the objective function in (5) then becomes [29]:

$$X(\mathbf{S}, \mathbf{C}, \mathbf{W}) = \sum_{l=1}^{k} \sum_{m=1}^{N} \sum_{A \in V} s_l w_{l,A}^{\beta} d(y_{m,A}, c_{l,A}) \tag{7}$$

subject to : $\sum_{A \in V} w_{l,A} = 1, 0 \leq w_{l,A} \leq 1$. The weight update formula for the formulation in (7) requires the computation of the intra-cluster distance from all pixels in an individual cluster to its cluster center in order to update the variable weights associated with that cluster. The main advantage of cluster-level variant is better boundary adherence in high-variance image areas, see Fig. 3.

The minimization of objective functions in (5) and (7) with respect to optimization variables **S**, **C**, and W (or **W**) is conducted using an iterative optimization strategy [27, 29]. Subproblems are defined on each optimization variable while keeping the other two variables constant. The respective solutions of subproblems on **S** and **C** are well known and each has a closed-form solution [27, 30].

3.2 AW*k*S algorithm

The AW*k*S segmentation is an iterative algorithm consisting of the following major steps : initialization, cluster assignment, and updates of cluster centers and the variable weights. Following [1, 25, 26, 31, 32], the input color image is first transformed into CIE-*Lab* color space to generate perceptual uniform superpixels [33].

Assignment step Similar to SLIC [1], AW*k*S algorithm selects a representative cluster for each image pixel i from among a few nearby candidate clusters $B_i \subset \mathbf{S}$ during assignment step. The set B_i for a pixel i contains all nearby clusters from **S** that accommodate pixel i in a square region of perimeter $\sqrt{16N/k}$ around their cluster centers [1]. As the distance computations for each pixel i has been restricted to the nearby cluster set $|B_i| << k$, a significant computation speedup is achieved over conventional k-means algorithm. In AW*k*S algorithm, an image pixel i is assigned to a nearby candidate cluster center C_l using the following simple rule: $Y_i \in S_l$ if $\sum_{A \in V} w_A^{\beta} d(y_{i,A}, c_{l,A}) \leq \sum_{A \in V} w_A^{\beta} d(y_{i,A}, c_{p,A})$ for $C_p \in C_B$ and $Y_i \notin S_p$ if $p \neq l$. If the distance measure is motivated by (7), we replace w_A^{β} by $w_{l,A}^{\beta}$ in the above rule.

Update step The assignment step generates a partitioning **S** of the input image where each pixel has a label from the set $\{1, 2, \dots, k\}$. The iterative AW*k*S optimization inspired from [27] performs separate update of both cluster centers, **C**, and the weights in the distance measure, W. During cluster center update, each lth cluster center in **C** is replaced by a v-D vector representing the mean of all pixels in **S** having the label l. An additional update step in W-k-means performs variable weights adjustment. The variable weights update procedure in AW*k*S is based on the current image partition **S**, the set of k-cluster centroids **C**, the chosen set V, and number of weights W to be updated. With **S** and **C** fixed, the update of W is carried out element-wise using the following equation [Theorem 1, [27]]:

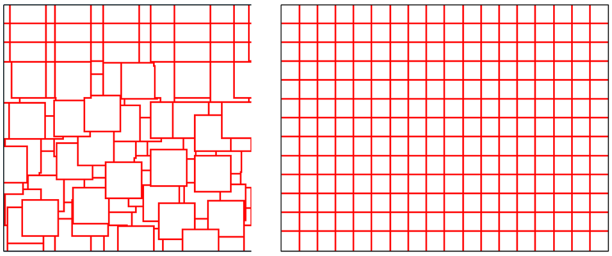
$$\hat{w}_A = \begin{cases} 0 & if\ D_A = 0 \\ \dfrac{1}{\sum_{\substack{F \in V, \\ s.t. D_F \neq 0}} \left[\frac{D_A}{D_F}\right]^{\frac{1}{\beta-1}}} & if\ D_A \neq 0 \end{cases} \tag{8}$$

iff $\beta > 1$ or $\beta \leq 0$. Further, D_A, as defined in (9), represents the variable-specific sum of intra-cluster distances between pixels in S_l and their respective cluster centers C_l.

$$D_A = \sum_{l=1}^{k} \sum_{m=1}^{N} s_l d(y_{m,A}, c_{l,A}) \tag{9}$$

The choice of values for β in (5) is particularly important and is justified in [27] on the basis of variable weighting principle. With $\beta < 0$ or $\beta > 1$, the weighting principle advocates a larger weight for a variable set A that has a smaller D_A (the sum of intra-cluster distances) and vice versa. In this way, insignificant variable sets $A \in V$ having large intra-cluster distances from the viewpoint of current clusters are assigned small weights, thereby reducing their effect in the subsequent clustering process. When $\beta = 0$, (5) loses the significance of weights and behaves identical to the conventional k-means. When β is set to 1, variable weighting becomes feature selection, thereby setting exactly one variable weight (corresponding to minimum D_A) to 1 from among all $A \in V$. A value of β chosen from the open set (0, 1) violets the weighting principle [27].

The direct use of (8) prefers variable selection over variable weighting when D_A is zero for some $A \in V$. Variable selection assigns a weight of zero to one or more variables, thereby eliminating their contribution in distance computations. In conventional SLIC with the CIE-*Lab* color-image plane feature space, we have V=$\{\{l, a, b\}, \{x, y\}\}$. Referring to Fig. 1, a constant intensity image should set $W_A = 0$ for $A = \{L, a, b\}$. But, due to the color space conversion from



(a) **(b)**

Fig. 1 Superpixel segmentation by *AWkS* on a constant intensity (white) image when the intra-cluster distances are computed **a** using Eq. 9 and **b** using Eq. 10. The thin black borders were added for clarity. They are not part of the data

RGB to CIE-Lab, D_A is set to an extremely small value in place of zero. Following (8) for weight computation, the weightage of W_{A^c} becomes extremely low which, therefore, produces false image superpixel segmentation as shown in Fig. 1a. To guard against this situation, the following modified equation has been used in AW*k*S for computing intra-cluster distances:

$$D_A = \sum_{l=1}^{k} \sum_{m=1}^{N} s_l d(y_{m,A}, c_{l,A}) + \zeta \tag{10}$$

where ζ should make the computation of D_A possible for all $A \in V$. To avoid variable rejection, a nonzero, positive constant value for ζ is sufficient [29]. However, for AW*k*S segmentation, ζ is set equal to the reciprocal of *m*th pixel variance in *Lab* color space. Let $\sigma^2_{lab}(m)$ denote the 3×3 local neighborhood variance around a pixel *m*, then $\zeta = 1/(\sigma^2_{lab}(m) + z)$, where *z* (set to .01) is a small constant. It is observed that (10) contributes to enforcing compactness in superpixels. More specifically, the very small local variance around a pixel in a flat image region improves (increases) the intra-cluster distance for $D_{\{L,a,b\}}$ as compared to $D_{\{x,y\}}$, when such distances are computed using (10). This results in assignment of relatively fair weight proportion to the spatial variable, and therefore, compactness is enforced. When a pixel is on some edge, the inter-cluster distance $D_{\{L,a,b\}}$ for the color space remains low compared to $D_{x,y}$, both due to the scale difference between spaces and a very small extra term being added to both intra-cluster distances. This scenario contributes in assigning fairly high weights to $W_{\{L,a,b\}}$, which produce superpixels that adhere well to image boundaries. The automatic update of the variable weights using (10) in each optimization step thus has advantages over user-specified constant terms. The solution $\hat{w}_{l,A}$ of (7) is similar to $\hat{w}_A$ in (8) with three important modifications [29]. First, the condition for $D_A = 0$ is removed. Second, D_A and D_F are replaced by $D_{l,A}$ and $D_{F,A}$, respectively, and finally, the intra-cluster distances are computed at cluster level using (11).

$$D_{l,A} = \sum_{m=1}^{N} s_l d(y_{m,A}, c_{l,A}) + \zeta \tag{11}$$

Weight Initialization The AW*k*S algorithm requires initialization of all weights $W(\mathbf{W})$ in (5) (7) in addition to the *k* cluster centers in **C**. The fact [27] that the final clustering result of AW*k*S algorithm depends on the initialization of both **C** and *W* rejects the possibility of initializing *W* randomly. Another possibility is to use the first run of conventional SLIC to obtain the initial image partition from which *W* can be initialized. Nevertheless, the resultant superpixels would be dependent on the specific choice of compactness parameter η and the algorithm would require two user-defined parameters. Boundary adaptive superpixel algorithms such as (VASLIC) [25] and nSLIC [26], on the other hand, require only one user-specified parameter with compactness controlled internally. This motivated us to utilize the clusters generated by exactly one iteration of VASLIC for fixed initialization of weights, $W(\mathbf{W})$ in (5), (7). VASLIC [25] incorporates the local variance of each image pixel (computed only on *L*-channel) into the distance measure as follows:

$$\begin{aligned} d_c &= \sqrt{\frac{\left(L_i - L_{C_l}\right)^2}{\left(\sigma^2_{\text{lab}}(i) + z\right)} + \left(a_i - a_{C_l}\right)^2 + (b_i - b_{C_l})^2} \\ D(i, C_l) &= \sqrt{d_c^2 + \left(\frac{d_s}{\sqrt{N/k}}\right)^2} \end{aligned} \tag{12}$$

where $\sigma^2_{\text{lab}}(i)$ is 3×3 local variance of pixel *i*, d_s is computed as in (2), and *z* is a small constant to avoid divide by zero. For further details on VASLIC, please refer to [25].

Termination The iterative optimization of **S**, **C** and *W* (or **W**) continues until the residual error $E = \|C^{\text{old}} - C^{\text{new}}\|_2$ converges, where the superscript on *C* refers to the previous and current cluster centroids in subsequent optimization steps. Most implementations and variants of SLIC have reported satisfactory results with iterations as few as 3 [26], 5 [25], and 10 [1]. In this work, all results are reported using 10 iteration of AW*k*S following conventional SLIC algorithm [1].

The clusters generated by AW*k*S algorithm may not have exactly one strongly connected component for each superpixel label. A post-processing step proposed in [1] remedies this shortcoming by assigning a quickly but randomly selected nearby label to each stray pixel set. Algorithm 1 summarizes the AW*k*S segmentation procedure.

Algorithm 1: Proposed Adaptive W-k-means based superpixel (AWkS) algorithm

```
Input: Image $I$ with $N$ pixels and desired number of superpixels, $k$.
// Initialization
Sample $k$ cluster centroids $C_k$ in the regularly spaced labxy
  color-image plane with grid interval $\sqrt{N/k}$.
Move cluster centers to the lowest gradient position in their $3 \times 3$
  neighborhood.
Set for each pixel $i$: label $l(i) := -1$, distance $d(i) := \infty$
Choose feature set $V$, and Initialize $W$ or $\mathbf{W}$ with 0.
Using $L$, compute $3 \times 3$ variance of each pixel $i$.
Set flag := 1
repeat
    // Label Assignment
    for each $C_l \in \boldsymbol{C}$ do
        for each pixel $i$ in a $2S \times 2S$ region around $C_l$ do
            if flag=1 then
                Compute $D(i, C_l)$ using (12)
            else
                if $W$ then
                    $D(i, C_l) = \sum_{A \in V} w_A^\beta d(y_{i,A}, c_{l,A})$
                else
                    $D(i, C_l) = \sum_{A \in V} w_{l,A}^\beta d(y_{i,A}, c_{l,A})$
                end
            end
            if $D(i, C_l) < d(i)$ then
                Set $d(i) := D(i, C_l)$
                Set $l(i) := l$
            end
        end
    end
    // Update steps
    for $l = 1 : k$ do
        for $n = 1 : v$ do
            $c_{l,n} = \sum_{m=1}^{N} s_l y_{m,n} / \sum_{m=1}^{N} s_l$
        end
    end
    Compute $E := \|C^{old} - C^{new}\|_2$
    if $E > threshold$ then
        Update $W$ using (10) or $\mathbf{W}$ using (11)
    end
    flag:=0
until $E \leq Threshold$
```

4 Experimental results

In this section, both qualitative and quantitative evaluation is performed to demonstrate the effectiveness of the AW*k*S segmentation algorithm under different distance measure configurations.

4.1 Evaluated configurations

Various distance measures discussed in Sect. 3 offer the flexibility to generate superpixels with different combinations of the feature sets *V* and the spatial extent of weighting in the feature space. Table 1 summarizes the various configurations of the AW*k*S segmentation algorithm. The "2-Feature" AW*k*S configurations perform automatic variable distance normalization for both variables which makes it different from SLIC. On the other hand, the "5-Feature" configurations reflect the relative importance of individual variables during cluster assignment. Both "2-Feature" and "5-Feature" cases are considered at image-level (I) and cluster-level (C) spatial weighting. For further discussion, the configurations are referred to by the names in the first column of Table 1. In this work, the performance of AW*k*S algorithm under four different configurations (distance measures) is evaluated against SLIC [1] and VASLIC [25].

4.2 Datasets

The AW*k*S segmentation configurations are evaluated on three challenging datasets. The Berkeley Segmentation Dataset 300 (BSDS300) [5] and its extension, BSDS500 [34], have been used in all recent works, viz. [1, 17, 25, 26, 31] for performance evaluation of superpixels algorithms. BSDS500 is an outdoor dataset with at least 5 hand-labeled, high-quality ground truth segmentations for each of the 500 images. Many images in BSDS500 are pictured in natural scenarios with unclear and hence hard to locate segment boundaries. The second dataset called the stanford background dataset (SBD) [35] is an outdoor dataset containing 715 images of varying size and visual quality. The presence of multiple foreground objects, objects with vague foreground boundaries, and the accommodation of detailed background regions in ground truth segmentations makes it challenging for the evaluation of superpixels. To evaluate AW*k*S segmentations, the third dataset used in this work is the Fashionista dataset (Fash) [36]. It contains 685 images showcasing fashion models in different outfits and poses before various backgrounds. The ground truth provided by [36] required preprocessing to meet the connected components requirements of superpixel segmentation [17]. Some test images from these datasets along with their pixel-accurate ground truth segmentations are shown in Fig. 2.

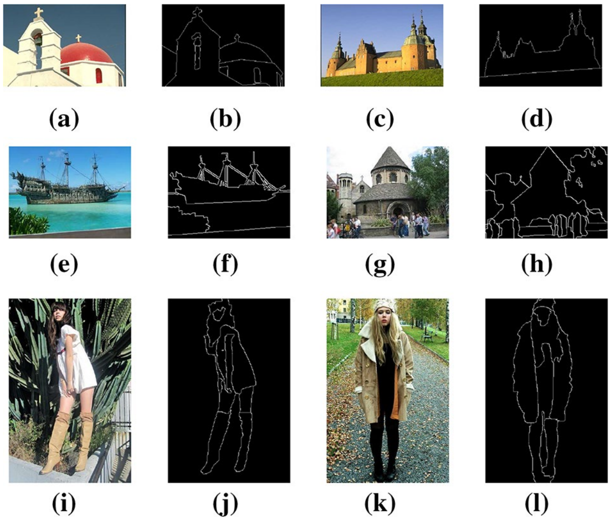

(a) (b) (c) (d)
(e) (f) (g) (h)
(i) (j) (k) (l)

Fig. 2 Some test images with their ground truth segmentation. BSDS500 dataset (Row 1), SBD dataset (Row 2), Fashionista dataset (Row 3). Computation of performance metrics such as boundary recall and undersegmentation error depends on pixel-accurate ground truth images for dataset images

4.3 Performance measures

Boundary recall (BR) [37] is a standard metric that measures the fraction of relevant edges that fall within at least two pixels of a retrieved boundary, where the pixels corresponding to relevant edges and retrieved boundary lie in the ground truth (**G**) and superpixels segmentation (**S**), respectively. A high boundary recall indicates better boundary adherence with very few relevant edges being missed [1].

Undersegmentation error (UE) [1, 38] is another standard metric that measures the "overspill" of superpixels with respect to ground truth **G**. It evaluates the boundary adherence based on the requirement that the spatial extent of a superpixel should not span more than one object in **G** [19]. Since there exists various formulations of UE as proposed by [1, 19, 38], our choice of UE in this work has the following formulation [38]:

$$\mathrm{UE}(G,S)=\frac{1}{N}\sum_{G_i\in G}\sum_{\substack{S_j\cap G_i\neq\phi,\\ S_j\in S}}\min\left\{S_j^{\mathrm{in}},S_j^{\mathrm{out}}\right\}, \tag{13}$$

where $S_j^{\mathrm{in}}=|S_j\cap G_i|$ and $S_j^{\mathrm{out}}=|S_j-G_i|$ represent the number of pixels of S_j that resides within and outside of G_i, respectively.

Table 1 Various AW*k*S configurations

Configuration	Variable set (V)	# of features(F)	Spatial extent	Equation to compute intra-cluster distance
AW*k*S2FI	$\{\{L, a, b\}, \{x, y\}\}$	2	Image	(10)
AW*k*S5FI	$\{L, a, b, x, y\}$	5	Image	(10)
AW*k*S2FC	$\{\{L, a, b\}, \{x, y\}\}$	2	Cluster	(11)
AW*k*S5FC	$\{L, a, b, x, y\}$	5	Cluster	(11)

Compactness proposed by [18] is a measure that works independently of the ground truth segments, G. This measure basically computes the sum over the iso-parametric quotients defined as $4\pi A_{S_j}/L_{S_j}^2$ for each superpixel S_j. Here, A_{S_j} and L_{S_j} are the area and length of S_j, respectively, each normalized by $\frac{|S_j|}{N}$. A higher iso-parametric quotient indicates better compactness.

Explained variance (EV) [39] utilizes the strong color and structure variations at image boundaries to measure the boundary adherence of superpixels. We use the formulation identical to [17, 39] in this work:

$$\text{EA}(S) = \frac{\sum_{S_j \in S} |S_j| \|\mu(S_j) - \mu(I)\|_2^2}{\sum_{y_n \in N} \|I(y_n) - \mu(I)\|_2^2}, \tag{14}$$

where $\mu(.)$ corresponds to the mean of an input argument and y_n is the nth pixel of image I having N total pixels. The judgment criteria for explained variance is higher is better.

Apart from boundary adherence and compactness, regularity and smoothness of superpixels are also important in visual quality inspection of superpixels [17]. In this context, superpixels that are nearly equal in size and systematic in the arrangement are termed regular. Smooth superpixels have boundaries that are expected to be free from artifacts.

4.4 Qualitative results

The visual quality of the superpixels generated by SLIC, VASLIC, and the AW*k*S configurations for some test images, drawn from all three datasets, is shown in Fig. 3. For all the implemented algorithms, the improvement in boundary adherence with an increase in desired superpixels is evident from Fig. 3. Although SLIC [1] superpixels do respect the image boundaries, they may still leave some salient edges in the image unattended, mainly due to the use of an adjusted distance function during k-means optimization. Some boundaries that are missed by the conventional SLIC are highlighted using blue rectangles under SLIC in Fig. 3. The VASLIC [25] algorithm performs by being greedy on image boundaries. It places many boundary points in the close vicinity of the actual edges rather than picking up the real boundary points at several places. Further, the presence of multiple boundaries near important edges contributes to high boundary adherence. As discussed in Sect. 3, all AW*k*S configurations have been initialized by a single run of VASLIC procedure [25]. Even with this initial setting, the difference in boundary adherence capabilities among the AW*k*S configurations and VASLIC is clearly observable. The AW*k*S segmentation under different distance measures has shown high boundary adherence capabilities akin to SLIC without being explicitly told about the compactness parameter. We highlight, using the blue rectangles, those important image boundaries that are captured by most of our implementations but are skipped by the conventional SLIC algorithm in Fig. 3.

In SLIC, a relatively high value of η with respect to $\sqrt{N/k}$ promotes compactness on the expense of boundary adherence and vice versa. With $\eta = 10$, all SLIC superpixels appear visually compact and regular in smooth image regions (see Fig 3). In contrast to this, VASLIC superpixels are highly non-compact, irregular, and non-smooth. VASLIC algorithm emphasizes boundary adherence over compactness as can be visually inspected in Fig. 3 and verified from the iso-parametric quotients plots for the three datasets in Fig. 4. The AW*k*S2FI and AW*k*S5FI superpixels outperform SLIC superpixels in terms of compactness on BSDS500 and FASH, and up to a large range of superpixel density on SBD dataset, please refer to Fig. 4. The high compactness of these configurations can be attributed to the modified equation for intra-cluster distances (10) as described in Sect.3. The rationale behind modifications in (10) and its counterpart for (7) is to improve the sum of intra-cluster distance for color features using pixels lying in the constant image areas. This contributes in setting comparatively low weight for color features making superpixels more compact. In detail image regions, the cluster-level computations in AW*k*S2FC and AW*k*S5FC set the intra-cluster distances for color features to relatively lower values compared to spatial features. Due to this, comparatively high weights are contributed to color features, thereby encouraging boundary adherence over compactness. The AW*k*S2FC and AW*k*S5FC superpixels therefore exhibit high boundary adherence at high-variance image areas, see Figs. 3 and 4. However, the compactness of AW*k*S2FC and AW*k*S5FC superpixels in smooth image regions is comparable to other AW*k*S configurations and SLIC (Fig. 3).

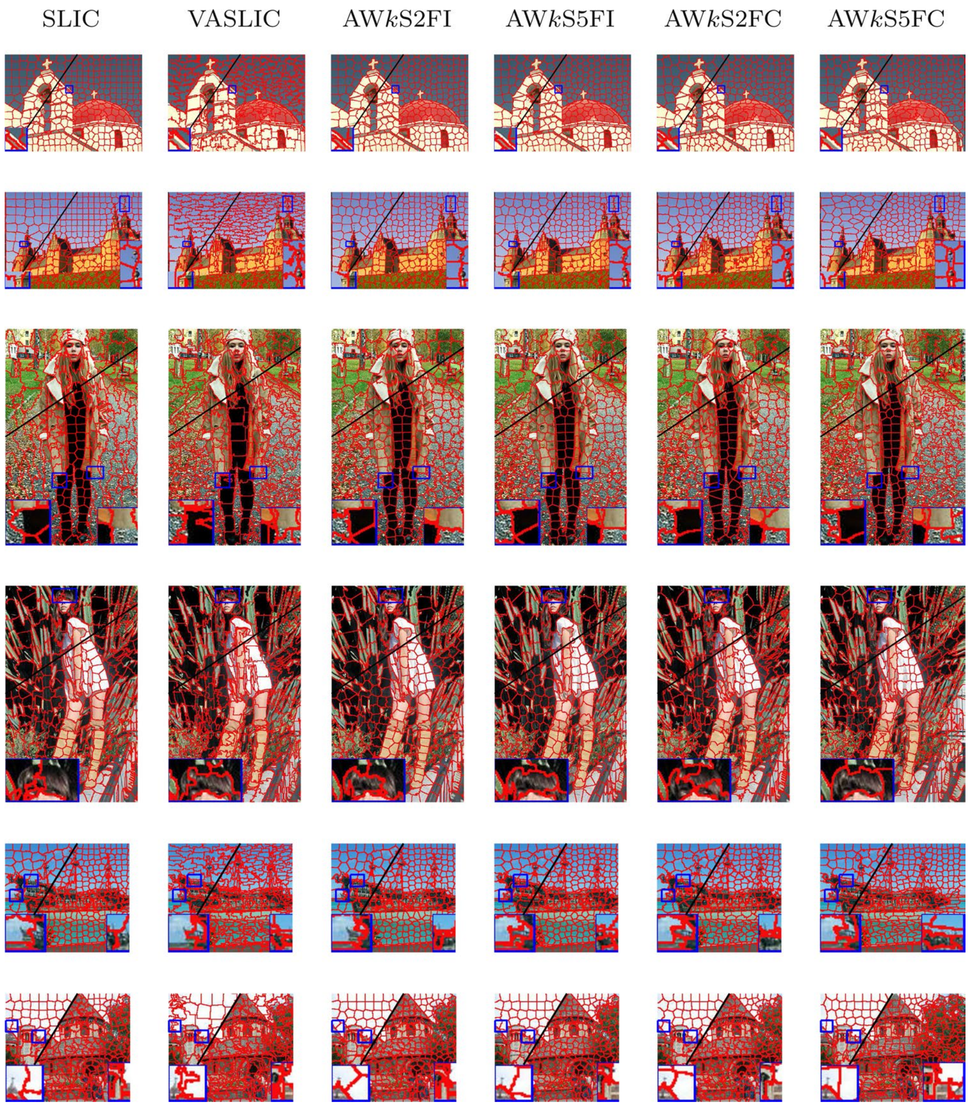


Fig. 3 Visual comparison among SLIC [1], VASLIC [25] and four configurations of AWkS algorithms. Test images belong to [row 1 and row 2] BSDS500 Dataset [34], [row 3 and row 4] FASH Dataset [36], and [row 5 and row 6] SBD Dataset [35]. Fragments forming the images above have $k \approx$ 250 in the upperleft corner and $k \approx$500 in the lower right corner

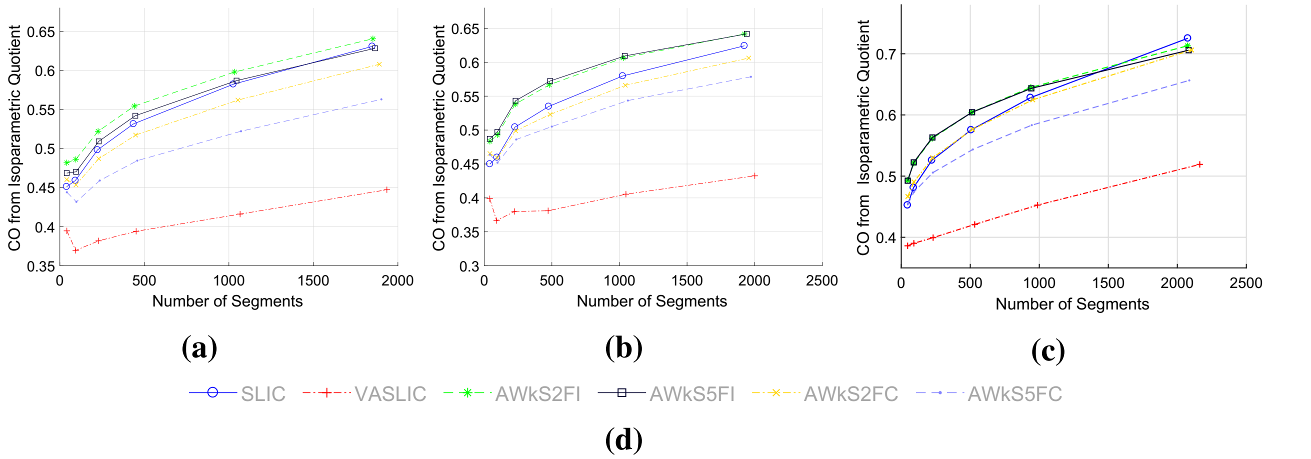


Fig. 4 Compactness plotted as a function of superpixel density on **a** BSDS500 dataset, **b** FASH dataset and **c** SBD dataset for SLIC, VASLIC and AWkS algorithm under different distance measures. **d** Legend

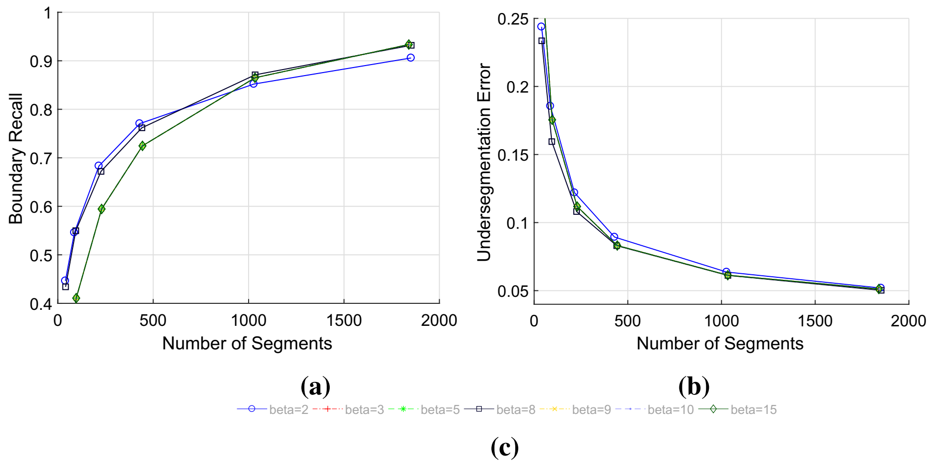
The induction of regularity and smoothness in SLIC superpixels while attempting compactness is not generally true for all superpixel algorithms [17]. However, the high regularity and smoothness of superpixels in our image-level configurations are eye-catching. It is observed that the use of fixed weights for all pixels during an assignment step could lead to compact, regular, and smooth superpixels. From these visual results, all AW*k*S configurations of superpixel algorithms appear to adhere well with the important object boundaries in most images, while maintaining compactness, regularity, and smoothness in flat image regions.

4.5 Quantitative results

In our experiments, all the AW*k*S configurations are executed with the desired number of primitives selected from the set $\{50, 100, 250, 500, 1000, 2000\}$. Various performance metrics such as boundary recall, undersegmentation error, explained variance, and compactness are computed for a given algorithm by their corresponding equations presented in Sect. 4.3 for each image in a dataset. The average across all images in a dataset for each metric is then plotted as a function of the number of (desired superpixel) segments. The term "superpixel density" is also used interchangeably to refer to the number of desired superpixel segments for images in a dataset. The choice of β influences the normalization of variables in (6). The value of β for different clustering datasets is recommended by [27] in the range $[4-9]$. To set the value of β in our experiments, we plot the boundary recall(BR) and undersegmentation error (UE) curves for AW*k*S2FI superpixels on the BSDS500 dataset with β selected from $\{2, 3, 5, 8, 9, 10, 15\}$. It can be observed from Fig. 5 that most values of β coincide and result in inferior superpixels than the ones for β in $\{2, 8\}$. The boundary adherence of superpixels with $\beta = 8$ is consistently better than other values of β on UE. Thus, the value of β is set to 8 for all our experiments.

The quantitative comparison of four AW*k*S configurations against SLIC and VASLIC is presented in Figs. 4 and 6. On the BSDS500 dataset, VASLIC superpixels achieve the highest BR performance, while the BR curves for all the AW*k*S configurations but AW*k*S2FI stay higher than the conventional SLIC algorithm. However, at high superpixel densities, all AW*k*S configurations outperform SLIC. VASLIC outperforms other methods on BR curves by choosing many boundary pixels in the vicinity of real image boundaries. In this way, it becomes the best by leveraging the flexibility offered by BR evaluation criteria. Interestingly, VASLIC does so on the cost of sacrificing on other quality measures such as compactness and smoothness in the entire superpixel density range, see Fig. 4a. As shown in Fig. 6b, the inability of VASLIC to avoid the "overspill" is indicated by its incredibly high UE curve. Further, the UE curves for all AW*k*S configurations are significantly better than such curves for SLIC and VASLIC. The UE curves for "5-Feature" configurations achieve lower values compared to "2-Feature" configurations. Please note that the BR of "5-feature" configuration was only next to the VASLIC in Fig. 6a. Further, AW*k*S2FI superpixels that were comparable to SLIC under BR perform a little better on the UE metric. In this regard, the observations made by Neburt and Protzel [38] that the boundary recall [5] is a measure that if used alone favors long boundaries and seeks attention. Further, as superpixels exhibit low precision, UE was asserted to be a more appropriate measure for boundary adherence than BR [38]. In terms of the explained variance, AW*k*S2FI and AW*k*S2FC perform better than the other configurations. When considered with UE, this indicates that these algorithms highlight points at boundary better than VASLIC. Under these observations, the boundary adherence performance of AW*k*S2FI, AW*k*S2FC, and AW*k*S5FI configurations is either better or comparable to SLIC [1].

Similar trends for UE and EV can be observed for Fashionista dataset, as shown in Fig. 6e, f. The ground truth

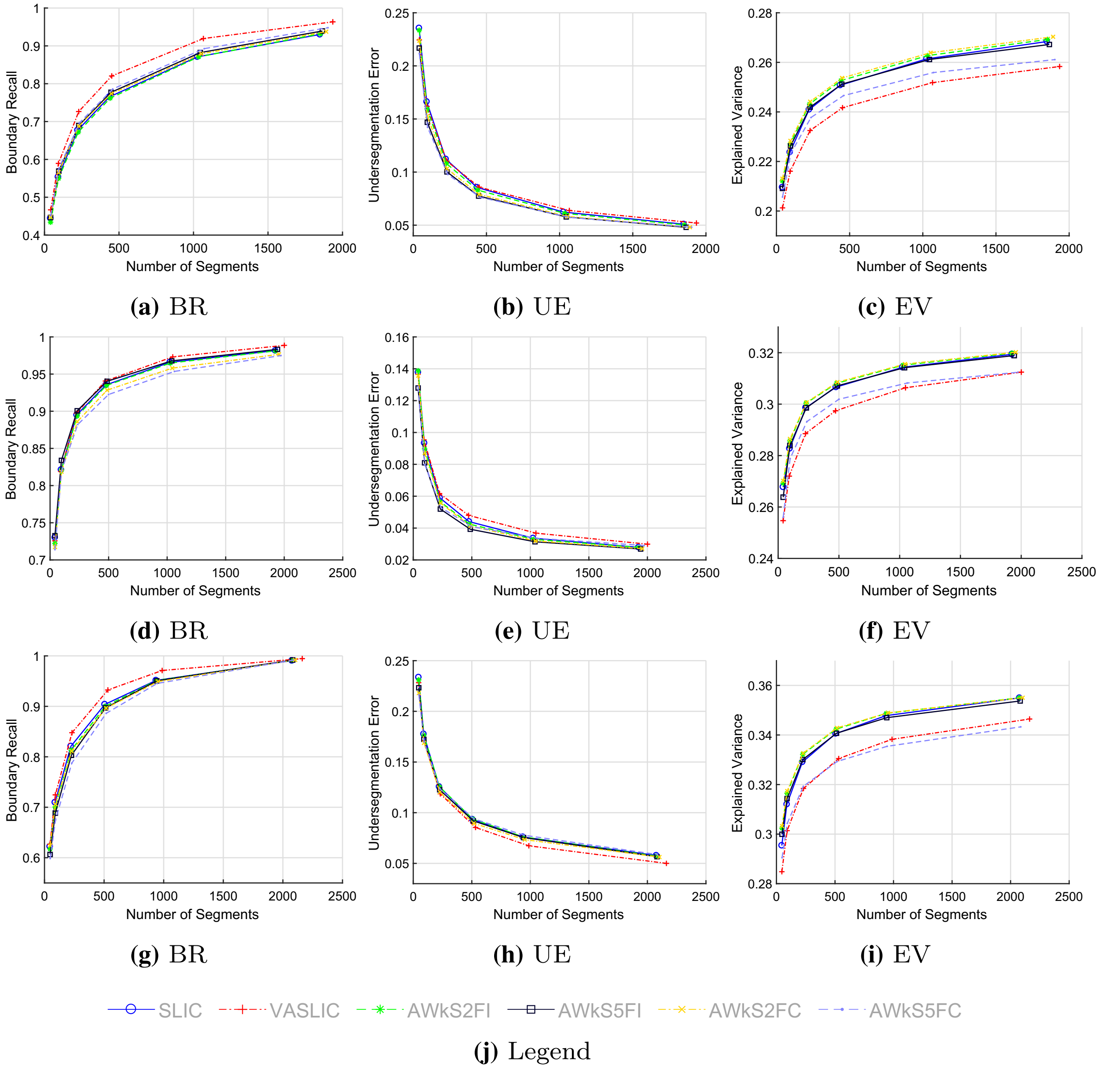


(a)

(b)

(c)

Fig. 5 Comparison of **a** boundary recall and **b** undersegmentation error on BSDS500 dataset for different values of β using AW*k*S2FI superpixels. Similar trends have been noticed for AW*k*S5FI superpixels as well. **c** Legend



(a) BR **(b)** UE **(c)** EV

(d) BR **(e)** UE **(f)** EV

(g) BR **(h)** UE **(i)** EV

(j) Legend

Fig. 6 Quantitative results on (1) BSDS500 dataset [first row: (**a**), (**b**), (**c**)], (2) Fashionista dataset [second row: (**d**), (**e**), (**f**)], (3) SBD dataset [third row: (**g**), (**h**), (**i**)]. Three boundary adherence performance measures: boundary recall (high is better), undersegmentation error (low is better), and explained variance (high is better), are plotted as a function of superpixel density for four configurations of AW*k*S, SLIC [1], and VASLIC [25]

segments in Fashionista datasets highlight only those boundaries that separate foreground models from the background. The BR, as shown in Fig. 6d, curves for FASH indicate that “2-feature” AW*k*S configurations obtain high adherence to object boundaries than “5-Features” for higher values of superpixels density. In the SBD dataset, the ground truth segmentation for images highlights all those boundary pixels that belong to either the most prominent foreground or background object in a scene. As the VASLIC is capable of capturing the boundaries of thin, non-compact regions, one can notice its curve becoming relatively better on both BR and UE for the SBD dataset. The high BR curve for SLIC in Fig. 6g indicates its ability in capturing the background objects better than the various AW*k*S configurations. However, SLIC could not avoid the “overspill” as can be observed from the UE curves for the SBD dataset in Fig. 6h. Except for AW*k*S5FC, all other configurations of AW*k*S algorithm perform either comparable or better than SLIC superpixels on UE and at lower superpixel densities on the EV metric for the SBD dataset.

The execution time of various configurations of AWkS is computed on the BSDS dataset with image size 481 X 321 on Intel Xeon CPU @ 2.7 GHz. The average time taken per image to generate 250 superpixels by AWkS2FI, AWkS5FI, AWkS2FC, and AWkS5FC is approx. 1.62, 3.24, 1.63, and 3.28 s, respectively. The average running time for the SLIC algorithm for the same dataset is 0.36 s. AWkS performs automatic weighting of the variables and therefore includes extra computations of intra-cluster distances which increases in the execution time. However, the parameterless and adaptive nature of AWkS saves the time spent in the manual tuning of parameters for optimal results.

5 Applications to saliency detection

The ability of the human vision system (HVS) to identify the perceptually attentive regions in an image without any prior contextual knowledge pertaining to the scene formed the basis for the field of saliency detection. The objective in the bottom-up saliency approach is to quickly identify the salient region(s) in an image using low, middle, or high level cues. Graphical methods such as [11–14, 28] under bottom-up saliency detection prefer to define a graph on superpixels (mid-level cues) for reducing computation load and extracting effective features.

Graphical models for saliency are expected to perform well when the constructed graph on superpixels is more interpretable. The AW*k*S superpixels follow the size constraints by being compact and regular in smooth image regions and thus have fewer neighbors in the induced graph. At detailed image regions improved boundary adherence for prominent objects allow the graph to encode shape information accurately. To test the AW*k*S superpixels for saliency detection, the preprocessing of input images in GBMR [14], RBD [28], and SMD [11] has been performed with the AW*k*S configurations. All these saliency detection approaches originally use SLIC in their preprocessing step. Please refer to the original papers for details on these saliency methods.

Experiments are conducted on SED single object dataset [40] and ECCSD multi-object dataset [41]. For each saliency algorithm, desired superpixels number is fixed. The F-measure curves are shown in Fig. 7. The mean average error (MAE), area under curve (AUC), overlap ratio (OR), and weighted F-measure (WF) scores are presented in Table 2. On single object SED dataset, GBMR with AW*k*S2FC superpixels achieves the best performance in terms of MAE, AUC, OR, and WF (Table 2A) as well as the F-measure curves (Fig. 7a) compared to all other superpixels. With RBD saliency detection, AW*k*S2FC performs the best in terms of AUC and WF, and the second in MAE and OR. In the F-measure curves in Fig. 7c, AW*k*S2FC performs the best at higher threshold ranges, while SLIC achieves the best at lower ranges. Finally, SMD with AW*k*S2FC obtains the best performance for all computed metrics and the best F-measure curve as reported in Table 2C and Fig. 7e, respectively. These results on SED dataset indicate that AW*k*S2FC

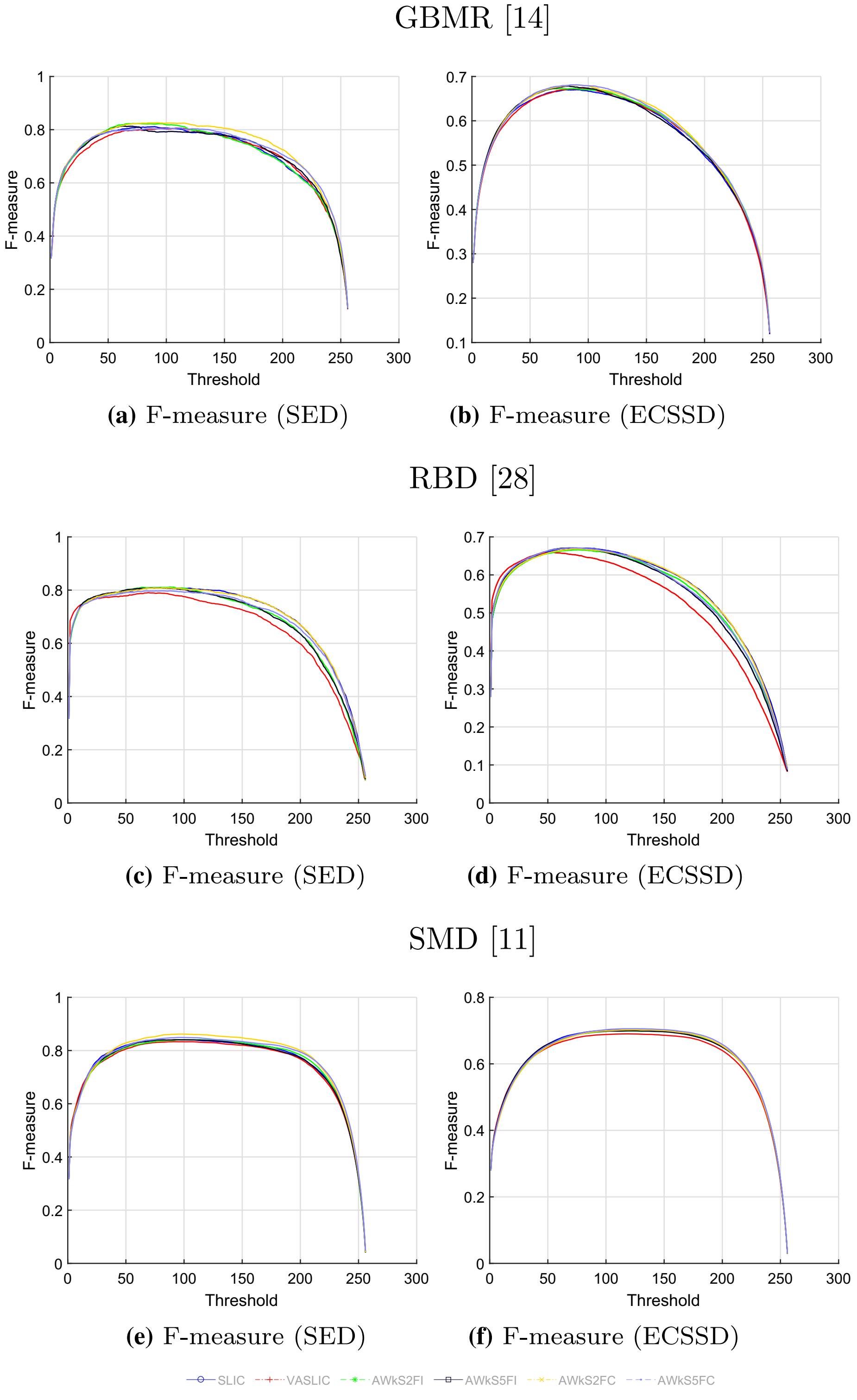
Table 2 Saliency detection results on SED and ECSSD datasets in terms of MAE, AUC, OR, and WF

SED							ECSSD					
Metric	SLIC	VASLIC	AW*k*S2FI	AW*k*S5FI	AW*k*S2FC	AW*k*S5FC	SLIC	VASLIC	AW*k*S2FI	AW*k*S5FI	AW*k*S2FC	AW*k*S5FC
(A) GBMR [14]												
MAE ↓	0.1500	0.1536	0.1477	0.1488	**0.1427**	*0.1473*	0.1901	0.1906	0.1884	*0.1871*	**0.1866**	0.1883
AUC ↑	0.8013	0.8069	0.8060	0.8069	**0.8086**	*0.8082*	0.7919	0.7929	0.7940	0.7945	*0.7969*	**0.7991**
OR ↑	0.6372	0.6325	0.6384	0.6422	**0.6544**	*0.6456*	0.5160	0.5180	0.5228	0.5261	*0.5283*	**0.5323**
WF ↑	0.6383	0.6251	0.6399	0.6395	**0.6534**	*0.6438*	0.4896	0.4890	0.4947	0.4923	*0.4957*	**0.4965**
(B) RBD [28]												
MAE ↓	**0.1405**	0.1539	0.1452	0.1458	*0.1419*	0.1451	*0.1713*	0.1741	0.1726	0.1718	0.1719	**0.1711**
AUC ↑	0.7927	0.7827	0.7921	*0.7948*	**0.7959**	0.7930	*0.7819*	0.7654	0.7787	0.7784	**0.7820**	0.7793
OR ↑	0.6249	0.6065	0.6265	**0.6317**	*0.6284*	0.6258	**0.5266**	0.5079	0.5229	*0.5255*	0.5248	0.5235
WF ↑	*0.6639*	0.6325	0.6538	0.6532	**0.6650**	0.6530	**0.5113**	0.4892	0.5043	0.5044	*0.5095*	0.5082
(C) SMD [11]												
MAE ↓	0.1246	0.1277	0.1256	0.1248	**0.1221**	*0.1234*	0.1741	0.1746	0.1746	**0.1730**	0.1763	*0.1741*
AUC ↑	0.8111	0.8100	0.8125	0.8138	**0.8208**	*0.8205*	0.8103	0.8027	0.8081	0.8091	*0.8110*	**0.8121**
OR ↑	0.6600	0.6535	0.6642	0.6670	**0.6782**	*0.6736*	0.5572	0.5475	0.5533	*0.5580*	0.5557	**0.5590**
WF ↑	0.6929	0.6886	0.6918	0.6927	**0.7065**	*0.7012*	0.5368	0.5316	0.5338	*0.5368*	0.5349	**0.5398**

↓ Smaller values are better. ↑ Larger values are better.

The best two results are highlighted with bold and italic, respectively

Fig. 7 F-measure curves. Under each saliency detection approach (GBMR, RBD, and SMD), curves **a**, **c** and **e** correspond to F-measures computed on SED single object dataset, whereas the curves shown in **b**, **d** and **f** correspond to F-measures computed on ECSSD dataset



(a) F-measure (SED) **(b)** F-measure (ECSSD)

(c) F-measure (SED) **(d)** F-measure (ECSSD)

(e) F-measure (SED) **(f)** F-measure (ECSSD)

(g)

superpixel outperforms SLIC [1] with saliency approaches such as GBMR [14] and SMD [11].

On ECSSD, GBMR with AW*k*S5FC scores best in terms of AUC, OR, and WF. GBMR with AW*k*S2FC performs the best for MAE and the second for AUC, OR, and WF. In the F-measure curve as shown in Fig. 7b, both GBMR with AW*k*S2FC and AW*k*S5FC are consistently superior to SLIC [1]. While RBD with cluster-level superpixels get the best score for MAE and AUC, SLIC superpixels-based RBD achieves the best in terms of OR and WF, and the second best in MAE and AUC. As evident from F-measure curves in Fig. 7d, RBD with AW*k*S2FC performs comparable to RBD with SLIC at higher threshold ranges. The high performance of SMD with AW*k*S5FC over SLIC is reflected in terms of the best values achieved for AUC, OR, and WF, and the F-measure in Fig. 7f. These results validate that the graph construction based on AW*k*S2FC and AW*k*S5FC superpixels is more interpretable and improves saliency detection performance in approaches such as GBMR [14] and SMD [11]. For detecting saliency in images with complex scenes, AW*k*S5FC superpixels perform better than AW*k*S2FC superpixels. This is also evident from the AW*k*S5FC BR (Fig. 6a) and UE (Fig. 6b) curves for the BSDS500 dataset.

6 Conclusion

In this work, an adaptive, W-*k*-means-based superpixel (AW*k*S) generation algorithm is proposed. Distance measures inspired by W-k-means clustering have been used to detect the amount of dis-similarity between the data points. Such distance formulations contain adaptive weighting terms for variables under consideration. Weights are assigned to each variable based on its relevance in the current partitioning of the image with respect to all variables. Based on feature sets and spatial extent of weight computation, four different distance measures under AW*k*S algorithm are evaluated. Superpixels generated by some of these AW*k*S configurations outperform SLIC superpixels on metrics like boundary adherence and compactness for datasets such as BSDS500 and FASH. All AW*k*S configurations are tested for saliency detection using three saliency detection approaches that include SLIC as a preprocessing step. The superiority of AW*k*S2FC and AW*k*S5FC superpixels over SLIC for saliency detection approaches like GBMR and SMD is observable from the various performance evaluation metrics.

Acknowledgements This work is partially supported by the project “Prediction of diseases through computer assisted diagnosis system using images captured by minimally invasive and non-invasive modalities,” Computer Science and Engineering, PDPM Indian Institute of Information Technology, Design and Manufacturing, Jabalpur, India (under ID: SPARC-MHRD-231). This work is partially supported by the SPEV project, University of Hradec Kralove, FIM, Czech Republic (ID: 2102–2020), “Smart Solutions in Ubiquitous Computing Environments.” We are also grateful for the support of students Sebastien Mambou and Michal Dobrovolny in consultations regarding application aspects.

References

1. Achanta R, Shaji A, Smith K, Lucchi A, Fua P, Süsstrunk S (2012) Slic superpixels compared to state-of-the-art superpixel methods. IEEE Trans Pattern Anal Mach Intell 34(11):2274–2282
2. Yan J, Yu Y, Zhu X, Lei Z, Li SZ (2015) Object detection by labeling superpixels. In: Proceedings of IEEE CVPR, pp 5107–5116
3. Mori G, Ren X, Efros AA, Malik J (2004) Recovering human body configurations: combining segmentation and recognition. In: Proceedings of IEEE CVPR, pp 326–333
4. He Y, Chiu WC, Keuper M, Fritz M, Campus SI (2017) Std2p: Rgbd semantic segmentation using spatio-temporal data-driven pooling. In: Proceedings of IEEE CVPR, pp 7158–7167
5. Ren X, Malik J (2003) Learning a classification model for segmentation. In: Proceedings of IEEE CVPR, pp 10–17
6. Lee T, Fidler S, Dickinson S (2015) Learning to combine mid-level cues for object proposal generation. In: Proceedings of IEEE ICCV, pp 1680–1688
7. Arbeláez P, Pont-Tuset J, Barron JT, Marques F, Malik J (2014) Multiscale combinatorial grouping. In: Proceedings of IEEE CVPR, pp 328–335
8. Wang S, Lu H, Yang F, Yang MH (2011) Superpixel tracking. In: Proceedings of IEEE ICCV, pp 1323–1330
9. Yang F, Lu H, Yang MH (2014) Robust superpixel tracking. IEEE Trans Image Process 23(4):1639–1651
10. Gupta S, Arbeláez P, Girshick R, Malik J (2015) Indoor scene understanding with rgb-d images: bottom-up segmentation, object detection and semantic segmentation. Int J Comput Vis 112(2):133–149
11. Peng H, Li B, Ling H, Hu W, Xiong W, Maybank SJ (2017) Salient object detection via structured matrix decomposition. IEEE Trans Pattern Anal Mach Intell 39(4):818–832
12. Sun X, He Z, Xu C, Zhang X, Zou W, Baciu G (2017) Diversity induced matrix decomposition model for salient object detection. Pattern Recognit 66:253–267
13. Jiang B, Zhang L, Lu H, Yang C, Yang MH (2013) Saliency detection via absorbing Markov chain. In: Proceedings of IEEE ICCV, pp 1665–1672
14. Yang C, Zhang L, Lu H, Ruan X, Yang MH (2013) Saliency detection via graph-based manifold ranking. In: Proceedings of IEEE CVPR, pp 3166–3173
15. Xiao F, Peng L, Lei F, Gao X (2018) Salient object detection based on eye tracking data. Signal Process 144:392–397
16. Zhang P, Zhuo T, Huang H, Kankanhalli M (2018) Saliency flow based video segmentation via motion guided contour refinement. Signal Process 142:431–440
17. Stutz D, Hermans A, Leibe B (2018) Superpixels: an evaluation of the state-of-the-art. Comput Vis Image Underst 166:1–27
18. Schick A, Fischer M, Stiefelhagen R (2012) Measuring and evaluating the compactness of superpixels. In: Proceedings of IEEE ICPR, pp 930–934
19. Liu MY, Tuzel O, Ramalingam S, Chellappa R (2011) Entropy rate superpixel segmentation. In: Proceedings of IEEE CVPR, pp 2097–2104
20. Vedaldi A, Soatto S (2008) Quick shift and kernel methods for mode seeking. In: Proceedings of IEEE ECCV, pp 705–718
21. Felzenszwalb PF, Huttenlocher DP (2004) Efficient graph-based image segmentation. Int J Comput Vis 59(2):167–181
22. Levinshtein A, Stere A, Kutulakos KN, Fleet DJ, Dickinson SJ, Siddiqi K (2009) Turbopixels: fast superpixels using geometric flows. IEEE Trans Pattern Anal Mach Intell 31(12):2290–2297
23. Neubert P, Protzel P (2014) Compact watershed and preemptive slic: On improving trade-offs of superpixel segmentation algorithms. In: Proceedings of IEEE ICPR, pp 996–1001
24. Gupta AK, Seal A, Khanna P (2019) Divergence based slic. Electron Lett 55(14):783–785
25. Kou F, Li Z, Wen C, Chen W (2016) Variance adaptive slic. In: Proceedings of IEEE ICIEA, pp 1671–1675
26. Jia S, Geng S, Gu Y, Yang J, Shi P, Qiao Y (2015) Nslic: slic superpixels based on nonstationarity measure. In: Proceedings of IEEE ICIP, pp 4738–4742

27. Huang JZ, Ng MK, Rong H, Li Z (2005) Automated variable weighting in k-means type clustering. IEEE Trans Pattern Anal Mach Intell 27(5):657–668
28. Zhu W, Liang S, Wei Y, Sun J (2014) Saliency optimization from robust background detection. In: Proceedings of IEEE CVPR, pp 2814–2821
29. Huang JZ, Xu J, Ng M, Ye Y (2008) Weighting method for feature selection in k-means. In: Computational methods of feature selection, pp 193–209
30. Selim SZ, Ismail MA (1984) K-means-type algorithms: a generalized convergence theorem and characterization of local optimality. IEEE Trans Pattern Anal Mach Intell 1:81–87
31. Choi KS, Oh KW (2016) Subsampling-based acceleration of simple linear iterative clustering for superpixel segmentation. Comput Vis Image Underst 146:1–8
32. Panigrahy C, Seal A, Mahato NK (2020) Fractal dimension of synthesized and natural color images in lab space. Pattern Anal Appl 23(2):819–836
33. Schwarz MW, Cowan WB, Beatty JC (1987) An experimental comparison of rgb, yiq, lab, hsv, and opponent color models. ACM Trans Graph 6(2):123–158
34. Arbelaez P, Maire M, Fowlkes C, Malik J (2011) Contour detection and hierarchical image segmentation. IEEE Trans Pattern Anal Mach Intell 33(5):898–916
35. Gould S, Fulton R, Koller D (2009) Decomposing a scene into geometric and semantically consistent regions. In: Proceedings of IEEE ICCV, pp 1–8
36. Yamaguchi K, Kiapour MH, Ortiz LE, Berg TL (2012) Parsing clothing in fashion photographs. In: Proceedings of IEEE CVPR, pp 3570–3577
37. Martin DR, Fowlkes CC, Malik J (2004) Contour detection and hierarchical image segmentation. IEEE Trans Pattern Anal Mach Intell 26(5):530–549
38. Neubert P, Protzel P (2012) Superpixel benchmark and comparison. In Proceedings of Forum Bildverarbeitung, p 6
39. Moore AP, Prince SJ, Warrell J, Mohammed U, Jones G (2008) Superpixel lattices. In: Proceedings of IEEE CVPR, pp 1–8
40. Alpert S, Galun M, Basri R, Brandt A (2007) Image segmentation by probabilistic bottom-up aggregation and cue integration. In: Proceedings of IEEE CVPR, pp 1–8
41. Yan Q, Xu L, Shi J, Jia J (2013) Hierarchical saliency detection. In: Proceedings of IEEE CVPR, pp 1155–1162

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.