Hierarchical Cellular Automata for Visual Saliency

Yao Qin* · Mengyang Feng* · Huchuan Lu · Garrison W. Cottrell

Received: date / Accepted: date

Abstract Saliency detection, finding the most important parts of an image, has become increasingly popular in computer vision. In this paper, we introduce Hierarchical Cellular Automata (HCA) – a temporally evolving model to intelligently detect salient objects. HCA consists of two main components: Single-layer Cellular Automata (SCA) and Cuboid Cellular Automata (CCA). As an unsupervised propagation mechanism, Single-layer Cellular Automata can exploit the intrinsic relevance of similar regions through interactions with neighbors. Low-level image features as well as high-level semantic information extracted from deep neural networks are incorporated into the SCA to measure the correlation between different image patches. With these hierarchical deep features, an impact factor matrix and a coherence matrix are constructed to balance the influences on each cell’s next state. The saliency values of all cells are iteratively updated according to a well-defined update rule. Furthermore, we propose CCA to integrate multiple saliency maps generated by SCA at different scales in a Bayesian framework. Therefore, single-layer propagation and multi-layer integration are jointly modeled in our unified HCA. Surprisingly, we find that the SCA can improve all existing methods that we applied it to, resulting in a similar precision level regardless of the original results. The CCA can act as an efficient pixel-wise aggregation algorithm that can integrate state-of-the-art methods, resulting in even better results. Extensive experiments on four challenging datasets demonstrate that the proposed algorithm outperforms state-of-the-art conventional methods and is competitive with deep learning based approaches.

Keywords Saliency Detection · Hierarchical Cellular Automata · Deep Contrast Features · Bayesian Framework

1 Introduction

Humans excel in identifying visually significant regions in a scene corresponding to salient objects. Given an image, people can quickly tell what attracts them most. In the field of computer vision, however, performing the same task is very challenging, despite dramatic progress in recent years. To mimic the human attention system, many researchers focus on developing computational models that locate regions of interest in the image. Since accurate saliency maps can assign relative importance to the visual contents in an image, saliency detection can be used as a pre-processing procedure to narrow the scope of visual processing and reduce the cost of computing resources. As a result, saliency detection has raised a great amount of attention (Achanta et al, 2009; Goferman et al, 2010) and has been incorporated into various computer vision tasks, such as visual tracking (Mahadevan and Vasconcelos, 2009), object retargeting (Ding et al, 2011; Sun and Ling, 2011) and image categorization (Siagian and Itti, 2007; Kanan and Cottrell, 2010). Results in perceptual research
show that contrast is one of the decisive factors in the human visual attention system (Itti and Koch, 2001; Reinagel and Zador, 1999), suggesting that salient objects are most likely in the region of the image that significantly differs from its surroundings. Many conventional saliency detection methods focus on exploiting local and global contrast based on various handcrafted image features, e.g., color features (Liu et al., 2011; Cheng et al., 2015), focusness (Jiang et al., 2013c), textual distinctiveness (Scharfenberger et al., 2013), and structure descriptors (Shi et al., 2013). Although these methods perform well on simple benchmarks, they may fail in some complex situations where the handcrafted low-level features do not help salient objects stand out from the background. For example, in Figure 1, the prairie dog is surrounded by low-contrast rocks and bushes. It is challenging to detect the prairie dog as a salient object with only low-level saliency cues. However, humans can easily recognize the prairie dog based on its category as it is semantically salient in high-level cognition and understanding.

In addition to the limitation of low-level features, the large variations in object scales also restrict the accuracy of saliency detection. An appropriate scale is of great importance in extracting the salient object from the background. One of the most popular ways to detect salient objects of different sizes is to construct multi-scale saliency maps and then aggregate them with predefined functions, such as averaging or a weighted summation. In most existing methods (Wang et al., 2016; Li and Yu, 2015; Li et al., 2014a; Zhou et al., 2014; Borji et al., 2015), however, these constructed saliency maps are usually integrated in a simple and heuristic way, which may directly limit the precision of saliency aggregation.

To address these two obvious problems, we propose a novel method named Hierarchical Cellular Automata (HCA) to extract the salient objects from the background efficiently. A Hierarchical Cellular Automata consists of two main components: Single-layer Cellular Automata (SCA) and Cuboid Cellular Automata (CCA). First, to improve the features, we use fully convolutional networks (Long et al., 2015) to extract deep features due to their successful application to semantic segmentation. It has been demonstrated that deep features are highly versatile and have stronger representational power than traditional handcrafted features (Krizhevsky et al., 2012; Farabet et al., 2013; Girshick et al., 2014). Low-level image features and high-level saliency cues extracted from deep neural networks are used by an SCA to measure the similarity of neighbors. With these hierarchical deep features, the SCA iteratively updates the saliency map through interactions with similar neighbors. Then the salient object will naturally emerge from the background with high consistency among similar image patches. Secondly, to detect multi-scale salient objects, we apply the SCA at different scales and integrate them with the CCA based on Bayesian inference. Through interactions with neighbors in a cuboid zone, the integrated saliency map can highlight the foreground and suppress the background.

An overview of our proposed HCA is shown in Figure 2. Furthermore, the Hierarchical Cellular Automata is capable of optimizing other saliency detection methods. If a saliency map generated by one of the existing methods is used as the prior map and fed into HCA, it can be improved to the state-of-the-art precision level. Meanwhile, if multiple saliency maps generated by different existing methods are used as initial inputs, HCA can naturally fuse these saliency maps and achieve a result that outperforms each method.

In summary, the main contributions of our work include:

1. We propose a novel Hierarchical Cellular Automata to adaptively detect salient objects of different scales based on hierarchical deep features. The model effectively improves all of the methods we have applied it to to state-of-the-art precision levels and is relatively insensitive to the original maps.

2. Single-layer Cellular Automata serve as a propagation mechanism that exploits the intrinsic relevance of similar regions via interactions with neighbors.

3. Cuboid Cellular Automata can integrate multiple saliency maps into a more favorable result under the Bayesian framework.
2 Related Work

2.1 Salient Object Detection

Methods of saliency detection can be divided into two categories: top-down (task-driven) methods and bottom-up (data-driven) methods. Approaches like (Alexe et al., 2010; Marchesotti et al., 2009; Ng et al., 2002; Yang and Yang, 2012) are typical top-down visual attention methods that require supervised learning with manually labeled ground truth. To better distinguish salient objects from the background, high-level category-specific information and supervised methods are incorporated to improve the accuracy of saliency maps. In contrast, bottom-up methods usually concentrate on low-level cues such as color, intensity, texture and orientation to construct saliency maps (Hou and Zhang, 2007; Jiang et al., 2011; Klein and Frintrop, 2011; Sun et al., 2012; Tong et al., 2015b; Yan et al., 2013). Some global bottom-up approaches tend to build saliency maps by calculating the holistic statistics on uniqueness of each element over the whole image (Cheng et al., 2015; Perazzi et al., 2012; Bruce and Tsotsos, 2005).

As saliency is defined as a particular part of an image that visually stands out compared to their neighboring regions or the rest of image, one of the most used principles, contrast prior, measures the saliency of a region according to the color contrast or geodesic distance against its surroundings (Cheng et al., 2013, 2015; Jiang et al., 2011; Jiang and Davis, 2013; Klein and Frintrop, 2011; Perazzi et al., 2012; Wang et al., 2011). Recently, the boundary prior has been introduced in several methods based on the assumption that regions along the image boundaries are more likely to be the background (Jiang et al., 2013b; Li et al., 2013; Wei et al., 2012; Yang et al., 2013; Borji et al., 2015; Shen and Wu, 2012), although this takes advantage of photographer’s bias and is less likely to be true for active robots. Considering the connectivity of regions in the background, Wei et al (2012) define the saliency value for each region as the shortest-path distance towards the boundary. Yang et al (2013) use manifold ranking to infer the saliency score of image regions according to their relevance to boundary superpixels. Furthermore, in (Jiang et al, 2013a), the contrast against the image border is used as a new regional feature vector to characterize the background.

However, one of the fundamental problems with all these conventional saliency detection methods is that the features used are not representative enough to capture the contrast between foreground and background, and this limits the precision of saliency detection. For one thing, low-level features cannot help salient objects stand out from a low-contrast background with similar visual appearance. Also, the extracted global features are weak in capturing semantic information and have much poorer generalization compared to the deep features used in this paper.

2.2 Deep Neural Networks

Deep convolutional neural networks have recently achieved a great success in various computer vision tasks, including image classification (Krizhevsky et al., 2012; Szegedy et al., 2015), object detection (Girshick et al., 2014; Hariharan et al., 2014; Szegedy et al., 2013) and semantic segmentation (Long et al., 2015; Pinheiro and Collobert, 2014). With the rapid development of deep neural networks, researchers have begun to construct effective neural networks for saliency detection (Zhao et al., 2015; Li and Yu, 2015; Zou and Komodakis, 2015; Wang et al., 2015; Li et al, 2016; Kim and Pavlovic, 2016). In (Zhao et al., 2015), Zhao et al. propose a unified multi-context deep neural network taking both...
global and local context into consideration. Li et al. (Li and Yu, 2015) and Zou et al. (Zou and Komodakis, 2015) explore high-quality visual features extracted from DNNs to improve the accuracy of saliency detection. DeepSaliency in (Li et al, 2016) is a multi-task deep neural network using a collaborative feature learning scheme between two correlated tasks, saliency detection and semantic segmentation, to learn better feature representation. One leading factor for the success of deep neural networks is the powerful expressibility and strong capacity of deep architectures that facilitate learning high-level features with semantic information (Hariharan et al, 2015; Ma et al, 2015).

In (Donahue et al, 2014), Donahue et al. point out that features extracted from the activation of a deep convolutional network can be repurposed to other generic tasks. Inspired by this idea, we use the hierarchical deep features extracted from fully convolutional networks (Long et al, 2015) to represent smaller image regions. The extracted deep features incorporate low-level features as well as high-level semantic information of the image and can be fed into our Hierarchical Cellular Automata to measure the similarity of different image patches.

2.3 Cellular Automata

Cellular Automata are a model of computation first proposed by Von Neumann (1951). They can be described as a temporally evolving system with simple construction but complex self-organizing behavior. A Cellular Automaton consists of a lattice of cells with discrete states, which evolve in discrete time steps according to specific rules. Each cell’s next state is determined by its current state as well as its nearest neighbors’ states. Cellular Automata have been applied to simulate the evolution of many complicated dynamical systems (Batty, 2007; Chopard and Droz, 2005; Cowburn and Welland, 2000; de Almeida et al, 2003; Martins, 2008; Pan et al, 2016). Considering that salient objects are spatially coherent, we introduce Cellular Automata into this field and propose Single-layer Cellular Automata as an unsupervised propagation mechanism to exploit the intrinsic relationships of neighboring elements of the saliency map and eliminate gaps between similar regions.

In addition, we propose a method to combine multi-scale saliency maps generated by different algorithms, or combine saliency maps at different scales through what we call Cuboid Cellular Automata (CCA). In CCA, states of the automaton are determined by a cuboid neighborhood corresponding to automata at the same location as well as their adjacent neighbors in different saliency maps. An illustration of the idea is in Figure 3(b). In this setting, the saliency maps are iteratively updated through interactions among neighbors in the cuboid zone. The state updates are determined through Bayesian evidence combination rules. Variants of this type of approach have been used before (Rahntu et al, 2010; Xie and Lu, 2011; Xie et al, 2013; Li et al, 2013). Xie et al (2013) use the low-level visual cues derived from a convex hull to compute the observation likelihood. Li et al (2013) construct saliency maps through dense and sparse reconstruction and propose a Bayesian algorithm to combine them. Using Bayesian updates to combine saliency maps puts the algorithm for Cuboid Cellular Automata on a firm theoretical foundation.

3 Proposed algorithm

In this paper, we propose an unsupervised Hierarchical Cellular Automata (HCA) for saliency detection, composed of two sub-units, a Single-layer Cellular Automata (SCA), and a Cuboid Cellular Automata (CCA), as described below. First, we construct prior maps of different scales with superpixels on the image boundary chosen as the background seeds. Then, hierarchical deep features are extracted from fully convolutional networks (Long et al, 2015) to measure the similarity of different superpixels. Next, we use SCA to iteratively update the prior maps at different scales based on the hierarchical deep features. Finally, a CCA is used to integrate the multi-scale saliency maps using Bayesian evidence combination. Figure 2 shows an overview of our proposed method.

3.1 Background Priors

Recently, there have been various mathematical models proposed to generate a coarse saliency map to help locate potential salient objects in an image (Tong et al, 2015a; Zhu et al, 2014; Gong et al, 2015). Even though prior maps are effective in improving detection precision, they still have several drawbacks. For example, a poor prior map may greatly limit the accuracy of the final saliency map if it incorrectly estimates the location of the objects or classifies the foreground as the background. Also, the computational time to construct a prior map can be excessive. Therefore, in this paper, we build a quite simple and time-efficient prior map that only provides the propagation seeds for HCA, which is quite insensitive to the prior map and is able to refine this coarse prior map into an improved saliency map.

First, we use the efficient Simple Linear Iterative Clustering (SLIC) algorithm (Achanta et al, 2010) to segment the image into smaller superpixels in order to capture the essential structural information of the image. Let \( s_i \in \mathbb{R} \) denote the saliency value of the superpixel \( i \) in the image. Based on the assumption that superpixels on the image boundary tend to have a higher
Hierarchical Cellular Automata for Visual Saliency

where \( df \) values, others, we assign a uniform value as their initial saliency to-zero saliency value to the boundary superpixels. For probability of being the background, we assign a close-

features of all contained pixels. The computed \( g \)

Each superpixel is represented by the mean of the deep

size via the nearest neighbor interpolation. Then each

we need to 1) crop the feature maps to get rid of the

about the features corresponding to the original image,

outputs of each convolutional layer in the FCN frame-

subsampling and pooling operations in the CNN, the

cannot precisely capture the fine-grained low-level in-

discriminate the objects from various backgrounds, they

As is well-known, the features in the last layers of CNNs

encode semantic abstractions of objects, and are robust to appearance variations, while the early layers contain low-level image features, such as color, edge, and texture. Although high-level features can effectively discriminate the objects from various backgrounds, they cannot precisely capture the fine-grained low-level information due to their low spatial resolution. Therefore, a combination of these deep features is preferred compared to any individual feature map.

In this paper, we use the feature maps extracted from the fully-convolutional network (FCN-32s (Long et al, 2015)) to encode object appearance. The input image to FCN-32s is resized to 500 \( \times \) 500, and a 100-pixel padding is added to the four boundaries. Due to subsampling and pooling operations in the CNN, the outputs of each convolutional layer in the FCN framework are not at the same resolution. Since we only care about the features corresponding to the original image, we need to 1) crop the feature maps to get rid of the padding; 2) resize each feature map to the input image size via the nearest neighbor interpolation. Then each feature map can be aggregated using a simple linear combination as:

\[
g(r_i, r_j) = \sum_{l=1}^{L} \rho_l \cdot \| df_i^l - df_j^l \|_2, \tag{2}\]

where \( df_i^l \) denotes the deep features of superpixel \( i \) on the \( l \)-th layer and \( \rho_l \) is a weighting of the importance of the \( l \)-th feature map, which we set by cross-validation. The weights are constrained to sum to 1: \( \sum_{l=1}^{L} \rho_l = 1. \) Each superpixel is represented by the mean of the deep features of all contained pixels. The computed \( g(r_i, r_j) \) is used to measure the similarity between superpixels.

3.3 Hierarchical Cellular Automata

Hierarchical Cellular Automata (HCA) is a unified framework composed of single-layer propagation (Single-layer Cellular Automata) and multi-layer aggregation (Cuboid Cellular Automata). It can generate saliency maps at different scales and integrate them to get a fine-grained saliency map. We will discuss SCA and CCA respectively in Sections 3.3.1 and 3.3.2.

3.3.1 Single-layer Cellular Automata

In Single-layer Cellular Automata (SCA), each cell denotes a superpixel generated by the SLIC algorithm. SLIC takes the number of desired superpixels as a parameter, so by using different numbers of superpixels with SCA, we can obtain maps at different scales. In this section, we assume one scale, denoted \( m \). We denote the number of superpixels in scale \( m \) by \( n_m \), but we omit the subscript \( m \) in most notation in this section for clarity, e.g., \( F \) for \( F_m \), \( C \) for \( C_m \) and \( s \) for \( s_m \).

We make three major modifications to the previous cellular automata models (Smith, 1972; Von Neumann, 1951) for saliency detection. First, the states of cells in most existing Cellular Automata models are discrete (Von Neumann et al, 1966; Wolfram, 1983). However, in this paper, we use the saliency value of each superpixel as its state, which is continuous between 0 and 1. Second, we give a broader definition of the neighborhood that is similar to the concept of \( z \)-layer neighborhood (here \( z = 2 \)) in graph theory. The \( z \)-layer neighborhood of a cell includes adjacent cells as well as those sharing common boundaries with its adjacent cells. Also, we assume that superpixels on the image boundaries are all connected to each other because all of them serve as background seeds. The connections between the neighbors are clearly illustrated in Figure 3 (a). Finally, instead of uniform influence of the neighbors, the influence is based on the similarity between the neighbor to the cell in feature space, as explained next.

**Impact Factor Matrix:** Intuitively, neighbors with more similar features have a greater influence on the

![Fig. 3](image-url)
cell’s next state. The similarity of any pair of superpixels is measured by a pre-defined distance in feature space. For the $m$-th saliency map, which has $n_m$ superpixels in total, we construct an impact factor matrix $F \in \mathbb{R}^{n_m \times n_m}$. Each element $f_{ij}$ in $F$ defines the impact factor of superpixel $i$ to $j$ as:

$$f_{ij} = \begin{cases} \exp\left(-\frac{g(r_i, r_j)}{\sigma_f}\right) & j \in \text{NB}(i) \\ 0 & j = i \text{ or otherwise}, \end{cases} \quad (3)$$

where $g(r_i, r_j)$ is a function that computes the distance between the superpixel $i$ and $j$ in feature space with $r_i$ as the feature descriptor of superpixel $i$. In this paper, we use the weighted distance of hierarchical deep features computed by Eqn. (2) to measure the similarity between neighbors. $\sigma_f$ is a parameter that controls the strength of similarity and NB$(i)$ is the set of the neighbors of the cell $i$. In order to normalize the impact factors, a degree matrix $D = \text{diag}\{d_1, d_2, \cdots, d_{n_m}\}$ is constructed, where $d_i = \sum_j f_{ij}$. Finally, a row-normalized impact factor matrix can be calculated as $F^* = D^{-1} F$.

Coherence Matrix: Given that each cell’s next state is determined by its current state as well as its neighbors, we need to balance the importance of these two factors. On the one hand, if a superpixel is quite different from all its neighbors in the feature space, its next state is determined by its current state as well as its state is determined by its current state as well as its neighbors. Second, it can be observed that there is a high contrast between the object and its surrounding background in feature space. Therefore, a clear boundary will naturally emerge between the object and the background, as the cell’s saliency value is greatly influenced by its similar neighbors. With boundary-based prior maps, salient objects can be naturally highlighted after the evolution of the system due to the connectivity and compactness of the object, as exemplified in Figure. 4. Specifically, even though part of the salient object is incorrectly selected as the background seed, the SCA can adaptively increase their saliency values under the influence of the local environment. The last three columns in Figure. 4 show that when the object touches the image boundary, the results achieved by the SCA are still satisfying.

Synchronous Update Rule: In Cellular Automata, all cells will simultaneously update their states according to the update rule, which is a key point in Cellular Automata, as it controls whether the ultimate evolving state is chaotic or stable (Wolfram, 1983). Here, we define the synchronous update rule based on the impact factor matrix $F^* \in \mathbb{R}^{n_m \times n_m}$ and coherence matrix $C^* \in \mathbb{R}^{n_m \times n_m}$:

$$s^{(t+1)} = C^* s^{(t)} + (I - C^*) F^* s^{(t)}, \quad (5)$$

where $I$ is the identity matrix of dimension $n_m \times n_m$ and $s^{(t)} \in \mathbb{R}^{n_m}$ denotes the saliency map at time $t$. When $t = 0$, $s^{(0)}$ is the prior map generated by the method introduced in Section. 3.1. After $T_S$ time steps (a time step is defined as one update of all cells), the saliency map can be represented as $s^{(T_S)}$. It should be noted that the update rule is invariant over time; only the cells’ states $s^{(t)}$ change over iterations.

Our synchronous update rule is based on the generalized intrinsic characteristics of most images. First, superpixels belonging to the foreground usually share similar feature representations. By exploiting the correlation between neighbors, the SCA can enhance saliency consistency among similar regions and develop a steady local environment. Second, it can be observed that there is a high contrast between the object and its surrounding background in feature space. Therefore, a clear boundary will naturally emerge between the object and the background, as the cell’s saliency value is greatly influenced by its similar neighbors.

3.3.2 Cuboid Cellular Automata

To better capture the salient objects of different scales, we propose a novel method named Cuboid Cellular Automata (CCA) to incorporate $M$ different saliency maps generated by SCA under $M$ scales, each of which serves as a layer of the Cuboid Cellular Automata. In CCA, each cell corresponds to a pixel, and the saliency values of all pixels constitute the set of cells’ states. The
number of all pixels in an image is denoted as \( H \). Unlike the
definition of a neighborhood in Section 3.3.1 and Multi-layer Cellular Automata in (Qin et al, 2015), here pixels
with the same coordinates in different saliency maps as well as their 4-connected pixels are all regarded
as neighbors. That is, for any cell in a saliency map, it should have 5\( M - 1 \) neighbors, constructing a cuboid
interaction zone. The hierarchical graph is presented in Figure. 3 (b) to illustrate the connections between
neighbors.

The saliency value of pixel \( i \) in the \( m \)-th saliency
map at time \( t \) is its probability of being the foreground
\( F \), represented as \( s_{m,i}^{(t)} = P(\eta_{m,j}^{(t)} = +1 | i \in m F) \), while \( 1 - s_{m,i}^{(t)} \) is
its probability of being the background \( B \), denoted as
\( 1 - s_{m,i}^{(t)} = P(\eta_{m,j}^{(t)} = B) \). We binarize each map with an
adaptive threshold using Otsu’s method (Otsu, 1975),
which is computed from the initial saliency map and does not change over time. The threshold of the \( m \)-th saliency map is denoted by \( \gamma_m \). If pixel \( i \) in the
\( m \)-th binary map is classified as foreground at time \( t \)
\( (s_{m,i}^{(t)} \geq \gamma_m) \), then it will be denoted as \( \eta_{m,i}^{(t)} = +1 \). Cor-
respondingly, \( \eta_{m,i}^{(t)} = -1 \) means that pixel \( i \) is binarized
as background \( (s_{m,i}^{(t)} < \gamma_m) \).

If pixel \( i \) belongs to the foreground, the probability
that one of its neighboring pixels \( j \) in the \( m \)-th binary
map is classified as foreground at time \( t \) \( (\eta_{m,j}^{(t)} = +1) \), then it
will be denoted as \( \lambda_{m,i}^{(t)} = +1 \). According to Eqn. (9) and Eqn. (10), we can get:
\[
\begin{align*}
\lambda_{m,i}^{(t)} &= P(\eta_{m,j}^{(t)} = +1 | i \in m F) \\
&= \frac{P(\eta_{m,j}^{(t)} = +1) - P(\eta_{m,j}^{(t)} = +1 | i \in m F)}{1 - \lambda_{m,i}^{(t)}} \\
&= s_{m,i}^{(t)} \frac{1 - \lambda_{m,i}^{(t)}}{1 - \lambda_{m,i}^{(t)}}.
\end{align*}
\]

Combining Eqn. (7) and Eqn. (8), the posterior ratio
\[\Omega(i \in m F | \eta_{m,j}^{(t)} = +1)\] takes into:
\[
\begin{align*}
\Omega(i \in m F | \eta_{m,j}^{(t)} = +1) &= \frac{P(i \in m F | \eta_{m,j}^{(t)} = +1)}{P(i \in m B | \eta_{m,j}^{(t)} = +1)} \\
&= \frac{s_{m,i}^{(t)}}{1 - s_{m,i}^{(t)}} \lambda.
\end{align*}
\]

As the posterior probability \( P(i \in m F | \eta_{m,j}^{(t)} = +1) \) represents the probability of pixel \( i \) of being the foreground \( F \) conditioned on that its neighboring pixel \( j \) in the \( m \)-th saliency map is binarized as foreground at time \( t \), \( P(i \in m F | \eta_{m,j}^{(t)} = +1) \) can also be used to rep-
resent the probability of pixel \( i \) of being the foreground
\( F \) at time \( t + 1 \). Then,
\[
\lambda_{m,i}^{(t+1)} = P(i \in m F | \eta_{m,j}^{(t)} = +1).
\]

According to Eqn. (9) and Eqn. (10), we can get:
\[
\lambda_{m,i}^{(t+1)} = \frac{P(i \in m F | \eta_{m,j}^{(t)} = +1)}{1 - P(i \in m B | \eta_{m,j}^{(t)} = +1)} \\
= \frac{P(i \in m F | \eta_{m,j}^{(t)} = +1)}{P(i \in m B | \eta_{m,j}^{(t)} = +1)} \\
= \frac{s_{m,i}^{(t)}}{1 - s_{m,i}^{(t)}} \frac{1 - \lambda_{m,i}^{(t)}}{1 - \lambda_{m,i}^{(t)}}.
\]

It is much easier to deal with the logarithm of this quan-
tity because the changes in logodds will be additive. So
Eqn. (11) turns into:
\[
\begin{align*}
\left( \Sigma_{m,i}^{(t+1)} \right) &= \ln \left( s_{m,i}^{(t+1)} \right) + \Lambda, \\
\Sigma_{m,i}^{(t)} &= \ln(\frac{1}{1 - s_{m,i}^{(t)}})
\end{align*}
\]

where \( \Lambda = \ln(\frac{1}{1 - \gamma_m}) \) is a constant. The intuitive explanation for Eqn. (12) is
that: if a pixel observes that one of its neighbors is bi-
inarized as foreground, it ought to increase its saliency
value; otherwise, it should decrease its saliency value.
Therefore, Eqn. (12) requires \( \Lambda > 0 \). In this paper, we
empirically set \( \Lambda = 0.05 \).

As each pixel has 5\( M - 1 \) neighbors in total, the pixel
will decide its action (increase or decrease it saliency
value) based on all its neighbors’ current states. Assum-
ing the contribution of each neighbor is conditionally
independent, we derive the synchronous update rule from
Eqn. (12) as:
\[
\begin{align*}
\left( \Sigma_{m,i}^{(t+1)} \right) &= \ln \left( s_{m,i}^{(t+1)} \right) + \Sigma_{m,i}^{(t)} .
\end{align*}
\]
where $s_{m,j}^{(t)} \in \mathbb{R}^H$ is the $m$-th saliency map at time $t$ and $H$ is the number of pixels in the image. $\Sigma_{m}^{(t)} \in \mathbb{R}^H$ can be computed by:

$$
\Sigma_{m}^{(t)} = \sum_{j=1}^{5} \sum_{k=1}^{M} \delta(k = m, j > 1) \cdot \text{sign} \left( s_{j,k}^{(t)} - \gamma_k \cdot 1 \right),
$$

(14)

where $M$ is the number of different saliency maps, $s_{j,k}^{(t)} \in \mathbb{R}^H$ is a vector containing the saliency values of the $j$-th neighbor for all the pixels in the $m$-th saliency map at time $t$ and $1 = [1, 1, \cdots, 1]^{\top} \in \mathbb{R}^H$. We use $\delta(k = m, j > 1)$ to represent the occasion that the cell only has 4 neighbors instead of 5 in the $m$-th saliency map when it is in the $m$-th saliency map. After $T_C$ iterations, the final integrated saliency map $s^{(T_C)}$ is calculated by

$$
s^{(T_C)} = \frac{1}{M} \sum_{m=1}^{M} s_{m}^{(T_C)}.
$$

(15)

In this paper, we use CCA to integrate saliency maps generated by SCA at $M = 3$ scales. This combination is denoted as HCA, and the visual saliency maps generated by HCA can be seen in Figure 5. Here we use the notation SCA$n$ to denote SCA applied with $n$ superpixels. We can see that the detected objects in the integrated saliency maps are uniformly highlighted and much closer to the ground truth.

3.4 Consistent Optimization

3.4.1 Single-layer Propagation

Due to the connectivity and compactness of the object, the salient part of an image will naturally emerge with the Single-layer Cellular Automaton, which serves as a propagation mechanism. Therefore, we use the saliency maps generated by several well-known methods as the prior maps and refresh them according to the synchronous update rule. The saliency maps achieved by CAS (Goferman et al, 2010), LR (Shen and Wu, 2012) and RC (Cheng et al, 2015) are taken as $s^{(0)}$ in Eqn. (13). The optimized results via SCA are shown in Figure 6. We can see that the foreground is uniformly highlighted and a clear object contour naturally emerges with the automatic single-layer propagation mechanism. Even though the original saliency maps are not particularly good, all of them are significantly improved to a similar accuracy level after evolution. That means our method is independent of prior maps and can make a consistent and efficient optimization towards state-of-the-art methods.

3.4.2 Pixel-wise Integration

A variety of methods have been developed for visual saliency detection, and each of them has its advantages and limitations. As shown in Figure 7, the performance of a saliency detection method varies with individual images. Each method can work well for some images or some parts of the images but none of them can perfectly handle all the images. Furthermore, different methods may complement each other. To take advantage of the superiority of each saliency map, we use Cuboid Cellular Automata to aggregate two groups of saliency maps, which are generated by three conventional algorithms: BL (Tong et al, 2015a), HS (Yan et al, 2013) and MR (Yang et al, 2013) and three deep learning methods: MDF (Li and Yu, 2015) and DS (Li et al, 2016) and MCDL (Zhao et al, 2015). Each of them serves as a layer of Cellular Automata $s_{m}^{(0)}$ in Eqn. (13). Figure 7 shows that our proposed pixel-wise aggregation method, Cuboid Cellular Automata, can appropriately integrate multiple saliency maps and outperforms each one. The saliency objects on the aggregated saliency map are consistently highlighted and much closer to the ground truth.
3.4.3 SCA + CCA = HCA

Here we show that when CCA is applied to some (poor) prior maps, it does not perform as well as when the prior map is post-processed by SCA. This motivates their combination into HCA. As is shown in Figure 8, when the candidate saliency maps are not well constructed, both CCA and MCA (Qin et al, 2015) fail to detect the salient object. Unlike CCA and MCA, HCA overcomes this limitation through incorporating single-layer propagation (SCA) together with pixel-wise integration (CCA) into a unified framework. The salient objects can be intelligently detected by HCA regardless of the original performance of the candidate methods. When we use HCA to integrate existing methods, the optimized results will be denoted as HCA*.

4 Experiments

In order to demonstrate the effectiveness of our proposed algorithms, we compare the results on four challenging datasets: ECSSD (Yan et al, 2013), MSRA5000 (Liu et al, 2011), PASCAL-S (Li et al, 2014b) and HKU-IS (Li and Yu, 2015). The Extended Complex Scene Saliency Dataset (ECSSD) contains 1000 images with multiple objects of different sizes. Some of the images come from the challenging Berkeley-300 dataset. MSRA-5000 contains more comprehensive images with complex background. The PASCAL-S dataset derives from the validation set of PASCAL VOC2010 (Everingham et al, 2010) segmentation challenge and contains 850 natural images. The last dataset, HKU-IS, contains 4447 challenging images and their pixel-wise saliency annotation. In this paper, we use ECSSD as the validation dataset to help choose the feature maps in FCN (Long et al, 2015).

We compare our algorithm with 20 classic or state-of-the-art methods including ITTI (Itti et al, 1998), FT (Achanta et al, 2009), CAS (Goferman et al, 2010), LR (Shen and Wu, 2012), XL13 (Xie et al, 2013), DSR (Li et al, 2013), HS (Yan et al, 2013), UFO (Jiang et al, 2013c), MR (Yang et al, 2013), DRFI (Jiang et al, 2013b), wCO (Zhu et al, 2014), RC (Cheng et al, 2015), HDCT (Kim et al, 2014), BL (Tong et al, 2015a), BSCA (Qin et al, 2015), LEGS (Wang et al, 2015), MCDL (Zhao et al, 2015), MDF (Li and Yu, 2015), DS (Li et al, 2016), and SSD-HS (Kim and Pavlovic, 2016), where the last 5 methods are deep learning-based methods. The results of different methods are either provided by authors or achieved by running available code or binaries. The code and results of HCA will be publicly available at our project site 1.

4.1 Parameter Setup

For the Single-layer Cellular Automaton, we set the number of iterations $T_S = 20$. $\sigma_f$ in Eq. (3) is set to 0.1 as in (Yang et al, 2013). For the Cuboid Cellular Automata, we set the number of iterations $T_C = 3$. We determined empirically that SCA and CCA converge by 20 and 3 iterations, respectively. We choose $M = 3$ and run SCA with $n_1 = 120, n_2 = 160, n_3 = 200$ superpixels to generate multi-scale saliency maps for CCA.

4.2 Evaluation Metrics

We evaluate all methods by standard Precision-Recall (PR) curves via binarizing the saliency map with a

---

1 [https://github.com/ArcherFMY/HCA_saliency_codes](https://github.com/ArcherFMY/HCA_saliency_codes)
In order to construct the Impact Factor matrix, we need to choose the features that will enter into Eqn. (2). Here we analyze the efficacy of the features in different layers of a deep network in order to choose those feature layers. In deep neural networks, earlier convolutional layers capture fine-grained low-level information, e.g., colors, edges and texture, while later layers capture high-level semantic features. In order to select the best feature layers in the FCN (Long et al., 2015), we use ECSSD as a validation dataset to measure the performance of deep features extracted from different layers. The function $g(r_i, r_j)$ in Eqn. (3) can be computed as

$$g(r_i, r_j) = \|df_i - df_j\|_2,$$

where $df_i$ denotes the deep features of superpixel $i$ on the $l$-th layer. The outputs of convolutional layers, relu layers and pooling layers are all regarded as a feature map. Therefore, we consider in total 31 layers of fully convolutional networks. We do not take the last two convolutional layers into consideration as their spatial resolutions are too low.

We use the F-measure (the higher, the better) and mean absolute error (MAE) (the lower, the better) to evaluate the performance of different layers on the ECSSD dataset. The results are shown in Figure. 10 (a) and (b). The F-measure score is obtained by thresholding the saliency maps at twice the mean saliency value. We use this convention for all of the subsequent F-measure results. The x-index in Figure. 10 (a) and (b) is set to 0.3 as suggested in (Achanta et al., 2009) to emphasize the precision. To complement these two measures, we also use mean absolute error (MAE) to quantitatively measure the average difference between the saliency map $s \in \mathbb{R}^H$ and the ground truth $g \in \mathbb{R}^H$ in pixel level:

$$\text{MAE} = \frac{1}{H} \sum_{i=1}^{H} |s_i - g_i|. \quad (18)$$

MAE indicates how similar a saliency map is compared to the ground truth, and is of great importance for different applications, such as image segmentation and cropping (Perazzi et al., 2012).

### 4.3 Validation of the Proposed Algorithm

#### 4.3.1 Feature Analysis

In order to construct the Impact Factor matrix, we need to choose the features that will enter into Eqn. (2). Here we analyze the efficacy of the features in different layers of a deep network in order to choose those feature layers. In deep neural networks, earlier convolutional layers capture fine-grained low-level information, e.g., colors, edges and texture, while later layers capture high-level semantic features. In order to select the best feature layers in the FCN (Long et al., 2015), we use ECSSD as a validation dataset to measure the performance of deep features extracted from different layers. The function $g(r_i, r_j)$ in Eqn. (3) can be computed as

$$g(r_i, r_j) = \|df_i - df_j\|_2,$$

where $df_i$ denotes the deep features of superpixel $i$ on the $l$-th layer. The outputs of convolutional layers, relu layers and pooling layers are all regarded as a feature map. Therefore, we consider in total 31 layers of fully convolutional networks. We do not take the last two convolutional layers into consideration as their spatial resolutions are too low.

We use the F-measure (the higher, the better) and mean absolute error (MAE) (the lower, the better) to evaluate the performance of different layers on the ECSSD dataset. The results are shown in Figure. 10 (a) and (b). The F-measure score is obtained by thresholding the saliency maps at twice the mean saliency value. We use this convention for all of the subsequent F-measure results. The x-index in Figure. 10 (a) and (b) is set to 0.3 as suggested in (Achanta et al., 2009) to emphasize the precision. To complement these two measures, we also use mean absolute error (MAE) to quantitatively measure the average difference between the saliency map $s \in \mathbb{R}^H$ and the ground truth $g \in \mathbb{R}^H$ in pixel level:

$$\text{MAE} = \frac{1}{H} \sum_{i=1}^{H} |s_i - g_i|. \quad (18)$$

MAE indicates how similar a saliency map is compared to the ground truth, and is of great importance for different applications, such as image segmentation and cropping (Perazzi et al., 2012).

### 4.3 Validation of the Proposed Algorithm

#### 4.3.1 Feature Analysis

In order to construct the Impact Factor matrix, we need to choose the features that will enter into Eqn. (2). Here we analyze the efficacy of the features in different layers of a deep network in order to choose those feature layers. In deep neural networks, earlier convolutional layers capture fine-grained low-level information, e.g., colors, edges and texture, while later layers capture high-level semantic features. In order to select the best feature layers in the FCN (Long et al., 2015), we use ECSSD as a validation dataset to measure the performance of deep features extracted from different layers. The function $g(r_i, r_j)$ in Eqn. (3) can be computed as

$$g(r_i, r_j) = \|df_i - df_j\|_2,$$

where $df_i$ denotes the deep features of superpixel $i$ on the $l$-th layer. The outputs of convolutional layers, relu layers and pooling layers are all regarded as a feature map. Therefore, we consider in total 31 layers of fully convolutional networks. We do not take the last two convolutional layers into consideration as their spatial resolutions are too low.

We use the F-measure (the higher, the better) and mean absolute error (MAE) (the lower, the better) to evaluate the performance of different layers on the ECSSD dataset. The results are shown in Figure. 10 (a) and (b). The F-measure score is obtained by thresholding the saliency maps at twice the mean saliency value. We use this convention for all of the subsequent F-measure results. The x-index in Figure. 10 (a) and (b) is set to 0.3 as suggested in (Achanta et al., 2009) to emphasize the precision. To complement these two measures, we also use mean absolute error (MAE) to quantitatively measure the average difference between the saliency map $s \in \mathbb{R}^H$ and the ground truth $g \in \mathbb{R}^H$ in pixel level:

$$\text{MAE} = \frac{1}{H} \sum_{i=1}^{H} |s_i - g_i|. \quad (18)$$

MAE indicates how similar a saliency map is compared to the ground truth, and is of great importance for different applications, such as image segmentation and cropping (Perazzi et al., 2012).

### 4.3 Validation of the Proposed Algorithm

#### 4.3.1 Feature Analysis

In order to construct the Impact Factor matrix, we need to choose the features that will enter into Eqn. (2). Here we analyze the efficacy of the features in different layers of a deep network in order to choose those feature layers. In deep neural networks, earlier convolutional layers capture fine-grained low-level information, e.g., colors, edges and texture, while later layers capture high-level semantic features. In order to select the best feature layers in the FCN (Long et al., 2015), we use ECSSD as a validation dataset to measure the performance of deep features extracted from different layers. The function $g(r_i, r_j)$ in Eqn. (3) can be computed as

$$g(r_i, r_j) = \|df_i - df_j\|_2,$$

where $df_i$ denotes the deep features of superpixel $i$ on the $l$-th layer. The outputs of convolutional layers, relu layers and pooling layers are all regarded as a feature map. Therefore, we consider in total 31 layers of fully convolutional networks. We do not take the last two convolutional layers into consideration as their spatial resolutions are too low.

We use the F-measure (the higher, the better) and mean absolute error (MAE) (the lower, the better) to evaluate the performance of different layers on the ECSSD dataset. The results are shown in Figure. 10 (a) and (b). The F-measure score is obtained by thresholding the saliency maps at twice the mean saliency value. We use this convention for all of the subsequent F-measure results. The x-index in Figure. 10 (a) and (b) is set to 0.3 as suggested in (Achanta et al., 2009) to emphasize the precision. To complement these two measures, we also use mean absolute error (MAE) to quantitatively measure the average difference between the saliency map $s \in \mathbb{R}^H$ and the ground truth $g \in \mathbb{R}^H$ in pixel level:

$$\text{MAE} = \frac{1}{H} \sum_{i=1}^{H} |s_i - g_i|. \quad (18)$$

MAE indicates how similar a saliency map is compared to the ground truth, and is of great importance for different applications, such as image segmentation and cropping (Perazzi et al., 2012).
apparent that salient objects are better detected with the final pooling layers of Conv1 and Conv5. Therefore, in this paper, we combine the feature maps from pool11 and pool15 with a simple linear combination. Eqn. (2) then turns into:

\[ g(r_i, r_j) = \rho_1 \cdot \|d_i^p - d_j^p\|_2 + \rho_2 \cdot \|d_i^{31} - d_j^{31}\|_2, \]

where \(\rho_1\) and \(\rho_2\) balance the weight of pool11 and pool15. In this paper, we empirically set \(\rho_1 = 0.375\) and \(\rho_2 = 0.625\) and apply them to all other datasets.

To test the effectiveness of the integrated deep features, we show the Precision-Recall curves of Single-layer Cellular Automata with each layer of deep features as well as the integrated deep features on two datasets. The Precision-Recall curves in Figure. 10 (c) demonstrate that hierarchical deep features outperform single-layer features, as they contain both category-level semantics and fine-grained details.

### 4.3.2 Component Effectiveness

To demonstrate the effectiveness of our proposed algorithm, we test the results on the standard ECSSD and PASCAL-S datasets. We generate saliency maps at three scales: \(n_1 = 120, n_2 = 160, n_3 = 200\) and use CCA to integrate them. FT curves in Figure. 11 indicate that the results of the Single-layer Automata are already quite satisfying. In addition, CCA can improve the overall performance of SCA with a wider range of high F-measure scores than SCA alone. Similar results are also achieved on other datasets but are not presented here to be succinct.

### 4.3.3 Performance Comparison

As is shown in Figure 12, our proposed Hierarchical Cellular Automata performs favorably against state-of-the-art conventional algorithms with higher precision and recall values on four challenging datasets. HCA is competitive with deep learning based approaches and has higher precision at low levels of recall. Furthermore, the fairly low MAE scores, displayed in Figure. 12(c), indicate that our saliency maps are very close to the ground truth. As MCDL (Zhao et al, 2015) trained the network on the MSRA dataset, we do not report its result on this dataset in Figure. 12. In addition, LEGS (Wang et al, 2015) used part of the images in the MSRA and PASCAL-S datasets as the training set. As a result, we only test LEGS with the test images on these two datasets. Saliency maps are shown in Figure. 16 for visual comparison of our method with other models.

### 4.4 Optimization of state-of-the-art methods

In the previous sections, we showed qualitatively that our model creates better saliency maps by improving initial saliency maps with SCA, or by combining the results of multiple algorithms with CCA, or by applying SCA and CCA. Here we compare our methods to other methods quantitatively. When the initial maps are imperfect, we apply SCA to improve them and then apply CCA. When the initial maps are already very good, we show that we can combine state-of-the-art methods to perform even better by simply using CCA.

### 4.4.1 Consistent Improvement

In Section 3.4.1, we concluded that results generated by different methods can be effectively optimized via Single-layer Cellular Automata. Figure 13 shows the precision-recall curves and mean absolute error bars of various saliency methods and their optimized results on two datasets. These results demonstrate that SCA can greatly improve existing results to a similar precision level. Even though the original saliency maps are not well constructed, the optimized results are comparable to the state-of-the-art methods. It should be noted that SCA can even optimize deep learning-based methods to a better precision level, e.g., MCDL (Zhao et al, 2015), MDF (Li and Yu, 2015), LEGS (Wang et al, 2015), and SSD-HS (Kim and Pavlovic, 2016). In addition, for one existing method, we can use SCA to optimize it at different scales and then use CCA to integrate the multi-scale saliency maps. The ultimate optimized result is denoted as HCA*. The lowest MAEs of saliency maps optimized by HCA in Figure 13 (c) show that HCA’s use of CCA improves performance over SCA alone.

### 4.4.2 Effective Integration

In Section 3.4.2, we used Cuboid Cellular Automata as a pixel-wise aggregation method to integrate two groups of state-of-the-art methods. One group includes three of the latest conventional methods while the other contains three deep learning-based methods. We test the
Fig. 12 PR curves, FT curves and MAE scores of different methods compared with our algorithm (HCA). From top to bottom: ECSSD, MSRA5000, PASCAL-S and HKU-IS are tested.
Fig. 13 Consistent improvement on ECSSD and PASCALS datasets. (a) and (b): PR curves of different methods (dashed line) and their optimized version via SCA200 (solid line). The right column shows that SCA200 (△), improves the MAEs of the original methods (×) and that HCA* (⊙), here applied to SCA120, SCA160, and SCA200, further improves the results.

Fig. 14 Effects of pixel-wise aggregation via Cuboid Cellular Automata on ECSSD dataset. The first row compares three conventional methods BL (Tong et al, 2015a), HS (Yan et al, 2013), MR (Yang et al, 2013) and their integrated results via Cuboid Cellular Automata, denoted as CCA. The second row compares three deep learning models, e.g. DS (Li et al, 2016), MCDL (Zhao et al, 2015), MDF (Li and Yu, 2015) and their integrated results. The precision, recall and F-measure scores in the right column are obtained by thresholding the saliency maps at twice the mean saliency value.
various methods on the ECSSD dataset, and the integrated result is denoted as CCA. PR curves in Figure 14(a) demonstrate the effectiveness of CCA over all the individual methods. FT curves of CCA in Figure 14(b) are fixed at high values that are insensitive to the thresholds. In addition, we binarize the saliency map with two times mean saliency value. From Figure 14(c) we can see that the integrated result has higher precision, recall and F-measure scores compared to each method that is integrated. Also, the mean absolute errors of CCA are always the lowest. The fairly low mean absolute errors indicate that the integrated results are quite similar to the ground truth.

Although Cuboid Cellular Automata have exhibited great strength in integrating multiple saliency maps, they have a major drawback in that the integrated result highly relies on the precision of the saliency detection methods used as input. If saliency maps fed into Cuboid Cellular Automata are not well constructed, it cannot naturally detect the salient objects via interactions between these candidate saliency maps. HCA, however, can easily address this problem by incorporating single-layer propagation and multi-layer integration into a unified framework. Unlike MCA (Qin et al, 2015) and CCA, HCA can achieve better integrated saliency map regardless of their original detection performance through the application of SCA to clean up the initial maps. PR curves, FT curves and MAE scores in Figure 15 show that 1) CCA has a better performance than MCA, as it considers the influence of adjacent cells on different layers. 2) HCA can greatly improve the aggregation results compared to MCA and CCA because it is independent of the initial saliency maps.

4.5 Run Time

The run time to process one image in the MSRA5000 dataset via Matlab R2014b-64bit with a PC equipped with an i7-4790k 3.60 GHz CPU and 32GB RAM is shown in Table 1. The Table displays the average run time of each component in our algorithm, not including the time for extracting deep features. We can see that the Single-layer Cellular Automata and Cuboid Cellular Automata are very fast at processing one image, on average 0.06s. Their combination HCA takes only 0.2421s
to process one image without superpixel segmentation and 1.0240s with SLIC.

We also compare the run time of our method with other state-of-the-art methods in Table 2. Here we compute the run time including superpixel segmentation and feature extraction for all models. Our algorithm has the least run time compared to other deep learning based methods and is the fifth fastest overall.

5 Conclusion

In this paper, we propose an unsupervised Hierarchical Cellular Automata, a temporally evolving system for saliency detection. It incorporates two components, Single-layer Cellular Automata (SCA), which can clean up noisy saliency maps, and Cuboid Cellular Automata (CCA), that can integrate multiple saliency maps. SCA is designed to exploit the intrinsic connectivity of saliency objects through interactions with neighbors. Low-level image features and high-level semantic information are both extracted from deep neural networks and incorporated into SCA to measure the similarity between neighbors. With superpixels on the image boundary chosen as the background seeds, SCA iteratively updates the saliency maps according to well-defined update rules, and salient objects naturally emerge under the influence of their neighbors. This context-based propagation mechanism can improve the saliency maps generated by existing methods to a high performance level. We used this in two ways: First, given a single saliency map, SCA can be applied to superpixels generated from the saliency map at multiple scales, and CCA can then integrate these into an improved saliency map. Second, we can take saliency maps generated by
multiple methods, apply SCA (if necessary) to improve them, and then apply CCA to integrate them into better saliency maps. Our experimental results demonstrate the superior performance of our algorithms compared to existing methods.

References

Achanta R, Hemami S, Estrada F, Susstrunk S (2009) Frequency-tuned salient region detection. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, pp 1597–1604

Achanta R, Shaji A, Smith K, Lucchi A, Fua P, Susstrunk S (2010) Slic superpixels. Tech. rep.

Alexe B, Deselaers T, Ferrari V (2010) What is an object? In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, pp 73–80

de Almeida CM, Batty M, Monteiro AMV, Câmara G, Soares-Filho BS, Cerqueira GC, Pennachin CL (2003) Stochastic cellular automata modeling of urban land use dynamics: empirical development and estimation. Computers, Environment and Urban Systems 27(5):481–509

Batty M (2007) Cities and complexity: understanding cities with cellular automata, agent-based models, and fractals. The MIT press

Borji A, Cheng MM, Jiang H, Li J (2015) Salient object detection: A benchmark. IEEE Transactions on Image Processing 24(12):5706–5722

Bruce N, Tsotsos J (2005) Saliency based on information maximization. In: Advances in neural information processing systems, pp 155–162

Cheng M, Mitra NJ, Huang X, Torr PH, Hu S (2015) Global contrast based salient region detection. IEEE Transactions on Pattern Analysis and Machine Intelligence 37(3):569–582

Cheng MM, Warrell J, Lin WY, Zheng S, Vineet V, Crook N (2013) Efficient salient region detection with soft image abstraction. In: Proceedings of the IEEE International Conference on Computer Vision, pp 1529–1536

Chopard B, Droz M (2005) Cellular automata modeling of physical systems, vol 6. Cambridge University Press

Cowburn R, Welland M (2000) Room temperature magnetic quantum cellular automata. Science 287(5457):1466

Ding Y, Xiao J, Yu J (2011) Importance filtering for image retargeting. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, pp 89–96

Donahue J, Jia Y, Vinyals O, Hoffman J, Zhang N, Tzeng E, Darrell T (2014) Decaf: A deep convolutional activation feature for generic visual recognition. In: Proceedings of International Conference on Machine Learning, pp 647–655

Everingham M, Van Gool L, Williams CK, Winn J, Zisserman A (2010) The pascal visual object classes (voc) challenge. International journal of computer vision 88(2):303–338

Farabet C, Couprie C, Najman L, LeCun Y (2013) Learning hierarchical features for scene labeling. IEEE transactions on pattern analysis and machine intelligence 35(8):1915–1929

Girshick R, Donahue J, Darrell T, Malik J (2014) Rich feature hierarchies for accurate object detection and semantic segmentation. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 580–587

Goferman S, Zelnik-manor L, Tal A (2010) Context-aware saliency detection. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition

Gong C, Tao D, Liu W, Maybank SJ, Fang M, Fu K, Yang J (2015) Saliency propagation from simple to difficult. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, pp 2531–2539

Hariharan B, Arbeláez P, Girshick R, Malik J (2014) Simultaneous detection and segmentation. In: Proceedings of European Conference on Computer Vision, Springer, pp 297–312

Hariharan B, Arbeláez P, Girshick R, Malik J (2015) Hypercolumns for object segmentation and fine-grained localization. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp 447–456

Hou X, Zhang L (2007) Saliency detection: A spectral residual approach. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, pp 1–8

Itti L, Koch C (2001) Computational modelling of visual attention. Nature reviews neuroscience 2(3):194–203

Itti L, Koch C, Niebur E (1998) A model of saliency-based visual attention for rapid scene analysis. IEEE Transactions on Pattern Analysis and Machine Intelligence (11):1254–1259

Jiang B, Zhang L, Lu H, Yang C, Yang MH (2013a) Saliency detection via absorbing markov chain. In: Proceedings of the IEEE International Conference on Computer Vision, pp 1665–1672

Jiang H, Wang J, Yuan Z, Liu T, Zheng N, Li S (2011) Automatic salient object segmentation based on context and shape prior. In: Proceedings of British Machine Vision Conference, vol 6, p 9
Hierarchical Cellular Automata for Visual Saliency

Jiang H, Wang J, Yuan Z, Wu Y, Zheng N, Li S (2013b) Salient object detection: A discriminative regional feature integration approach. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, pp 2083–2090

Jiang P, Ling H, Yu J, Peng J (2013c) Salient region detection by UFO: Uniqueness, focusness and objectness. In: Proceedings of the IEEE International Conference on Computer Vision, pp 1976–1983

Jiang Z, Davis L (2013) Submodular salient region detection. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, pp 2472–2479

Kim J, Pavlovic V (2016) A shape-based approach for salient object detection using deep learning. In: Proceedings of European Conference on Computer Vision, pp 455–470

Kim J, Han D, Tai YW, Kim J (2014) Salient region detection via high-dimensional color transform. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp 883–890

Klein DA, Frintrop S (2011) Center-surround divergence of feature statistics for salient object detection. In: Proceedings of the IEEE International Conference on Computer Vision, pp 2214–2219

Krizhevsky A, Sutskever I, Hinton GE (2012) Imagenet classification with deep convolutional neural networks. In: Advances in neural information processing systems, pp 1097–1105

Li G, Yu Y (2015) Visual saliency based on multiscale deep features. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, pp 5455–5463

Li N, Ye J, Ji Y, Ling H, Yu J (2014a) Saliency detection on light field. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition

Li X, Lu H, Zhang L, Ruan X, Yang MH (2013) Saliency detection via dense and sparse reconstruction. In: Proceedings of the IEEE International Conference on Computer Vision, pp 2976–2983

Li X, Zhao L, Wei L, Yang MH, Wu F, Zhuang Y, Ling H, Wang J (2016) Deep saliency: Multi-task deep neural network model for salient object detection. IEEE Transactions on Image Processing 25(8):3919–3930

Li Y, Hou X, Koch C, Rehg J, Yuille A (2014b) The secrets of salient object segmentation. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, pp 280–287

Liu T, Yuan Z, Sun J, Wang J, Zheng N, Tang X, Shum HY (2011) Learning to detect a salient object. IEEE Transactions on Pattern Analysis and Machine Intelligence 33(2):353–367

Long J, Shelhamer E, Darrell T (2015) Fully convolutional networks for semantic segmentation. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, pp 3431–3440

Ma C, Huang JB, Yang X, Yang MH (2015) Hierarchical convolutional features for visual tracking. In: Proceedings of the IEEE International Conference on Computer Vision, pp 3074–3082

Mahadevan V, Vasconcelos N (2009) Saliency-based discriminant tracking. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, pp 1007–1013

Marchesotti L, Cifarelli C, Csurka G (2009) A framework for visual saliency detection with applications to image thumbnailing. In: Proceedings of the IEEE International Conference on Computer Vision, pp 2232–2239

Martins AC (2008) Continuous opinions and discrete actions in opinion dynamics problems. International Journal of Modern Physics C 19(04):617–624

Ng AY, Jordan MI, Weiss Y, et al (2002) On spectral clustering: Analysis and an algorithm. Advances in neural information processing systems 2:849–856

Otsu N (1975) A threshold selection method from gray-level histograms. Automatica 11(285-296):23–27

Pan Q, Qin Y, Xu Y, Tong M, He M (2016) Opinion evolution in open community. International Journal of Modern Physics C p 1750003

Perazzi F, Križenbühl P, Pritch Y, Hornung A (2012) Saliency filters: Contrast based filtering for salient region detection. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, IEEE, pp 733–740

Pinheiro PH, Collobert R (2014) Recurrent convolutional neural networks for scene labeling. In: Proceedings of International Conference on Machine Learning, pp 82–90

Qin Y, Lu H, Xu Y, Wang H (2015) Saliency detection via cellular automata. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition

Reinagel P, Zador AM (1999) Natural scene statistics at the centre of gaze. Network: Computation in Neural Systems 10(4):341–350

Scharfenberger C, Wong A, Fergani K, Zelek JS, Clausi DA (2013) Statistical textural distinctiveness for salient region detection in natural images. In: Proceedings of IEEE Conference on Computer Vision
and Pattern Recognition, pp 979–986
Shen X, Wu Y (2012) A unified approach to salient object detection via low rank matrix recovery. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, pp 853–860
Shi K, Wang K, Lu J, Lin L (2013) Pisa: Pixelwise image saliency by aggregating complementary appearance contrast measures with spatial priors. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, pp 2115–2122
Siagian C, Itti L (2007) Rapid biologically-inspired scene classification using features shared with visual attention. IEEE Transactions on Pattern Analysis and Machine Intelligence 29(2):300–312
Smith AR (1972) Real-time language recognition by one-dimensional cellular automata. Journal of Computer and System Sciences 6(3):233–253
Sun J, Ling H (2011) Scale and object aware image retargeting for thumbnail browsing. In: Proceedings of the IEEE International Conference on Computer Vision, pp 1511–1518
Sun J, Lu H, Li S (2012) Saliency detection based on integration of boundary and soft-segmentation. In: Proceedings of IEEE International Conference on Image Processing, pp 1085–1088
Szegedy C, Toshev A, Erhan D (2013) Deep neural networks for object detection. In: Advances in Neural Information Processing Systems, pp 2553–2561
Szegedy C, Liu W, Jia Y, Sermanet P, Reed S, Anguelov D, Erhan D, Vanhoucke V, Rabinovich A (2015) Going deeper with convolutions. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp 1–9
Tong N, Lu H, Ruan X, Yang MH (2015a) Salient object detection via bootstrap learning. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp 1884–1892
Tong N, Lu H, Zhang Y, Ruan X (2015b) Salient object detection via global and local cues. Pattern Recognition 48(10):3258–3267
Von Neumann J (1951) The general and logical theory of automata. Cerebral mechanisms in behavior 1(41):1–2
Von Neumann J, Burks AW, et al (1966) Theory of self-reproducing automata. IEEE Transactions on Neural Networks 5(1):3–14
Wang L, Xue J, Zheng N, Hua G (2011) Automatic salient object extraction with contextual cue. In: Proceedings of the IEEE International Conference on Computer Vision, pp 105–112
Wang L, Lu H, Ruan X, Yang MH (2015) Deep networks for saliency detection via local estimation and global search. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, pp 3183–3192
Wang Q, Zheng W, Piramuthu R (2016) Grab: Visual saliency via novel graph model and background priors. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, pp 535–543
Wei Y, Wen F, Zhu W, Sun J (2012) Geodesic saliency using background priors. In: Proceedings of European Conference on Computer Vision, pp 29–42
Wolfram S (1983) Statistical mechanics of cellular automata. Reviews of modern physics 55(3):601
Xie Y, Lu H (2011) Visual saliency detection based on bayesian model. In: Proceedings of IEEE International Conference on Image Processing, pp 645–648
Xie Y, Lu H, Yang MH (2013) Bayesian saliency via low and mid level cues. IEEE Transactions on Image Processing 22(5):1689–1698
Yan Q, Xu L, Shi J, Jia J (2013) Hierarchical saliency detection. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, pp 1155–1162
Yang C, Zhang L, Lu H, Ruan X, Yang MH (2013) Saliency detection via graph-based manifold ranking. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, pp 3166–3173
Yang J, Yang MH (2012) Top-down visual saliency via joint crf and dictionary learning. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, IEEE, pp 2296–2303
Zhao R, Ouyang W, Li H, Wang X (2015) Saliency detection by multi-context deep learning. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, pp 1265–1274
Zhou F, Bing Kang S, Cohen MF (2014) Time-mapping using space-time saliency. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition
Zhu W, Liang S, Wei Y, Sun J (2014) Saliency optimization from robust background detection. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, pp 2814–2821
Zou W, Komodakis N (2015) Harf: Hierarchy-associated rich features for salient object detection. In: Proceedings of the IEEE International Conference on Computer Vision, pp 406–414