Geodescially Driven Deep Level Sets for Salient Objects Detection

Mohammadjavad Matinkia, Reza Safabakhsh

Abstract—The power of deep neural networks in extracting high level features from an image has made them a proper choice for integrating them with classic computer vision methods, such as active contour models and level set methods. Active contour models are based on minimizing an energy functional which leads to an evolving partial differential equation (PDE). The steady state solution of the PDE, ideally represents the boundary of a foreground object. In this paper, we propose a novel method to train a convolutional neural network based on the concept of the level set method. The proposed method is supported by a concrete geometrical intuition and proof. More precisely, we model the boundary of the foreground objects as the geodesics on a three-dimensional regular surface, and derive a local condition for a planar curve to be a geodesic on a regular surface. Using this condition, we develop a novel cost function to train a variety of convolutional neural networks. We test our model on five different datasets, one of which containing camouflaged objects (**CAMO** dataset), and demonstrate that our model achieves the state-of-the-art performance in salient object detection.

Index Terms—Salient object detection, Regular surfaces, Geodesics, Active Contour Models, Level Set method, Deep convolutional networks

1 Introduction

Object detection has been one of the most fundamental problems in computer vision due to its theoretical significance and applications, [1], [2], [3], [4]. *Salient object detection* as a category of object detection problems is defined as finding the most salient object in the image. In other words, in salient object detection, one tries to produce binary images from an input image in which the pixels corresponding to the foreground or target object have the value 255 (white) and the pixels corresponding to the background have the value 0 (black). In classical computer vision, one of the methods for finding objects is through object boundary detection. Among various methods of edge detection, *Active Contour Models* are one the most attractive and efficacious methods [5], [6], [7]. These methods strive to find the boundary of an object by minimizing an energy functional [8]. In these methods, the boundary of the object is modeled either as an evolving curve on $\mathbb{R}^2$ plane whose energy is minimized when enveloping the region of interest (Snake model) or as the intersection of an evolving three-dimensional surface and the $\mathbb{R}^2$ plane whose evolution PDE yields to an intersection which agrees with the boundary of the target (Level Set method). In modern computer vision, solving the problem of object detection and salient object detection has been facilitated by using *deep neural networks* [9], [10], [11], [12], [13]. The deep features extracted from the neural networks have been shown to be more powerful and effective than the low level features extracted in classical computer vision [14]. However, the methods from classical computer vision could still remain useful in the sense of prior knowledge for the neural networks or designing better loss functions.

In this paper, we intend to combine the *Level Set* method from the classical computer vision and *Convolutional Neural Networks* to introduce a novel cost function for training convolutional neural networks. We propose a geometrical approach for training the neural network to find the salient objects. More precisely, we try to find three-dimensional regular surfaces with proper global geometric properties whose intersection with $\mathbb{R}^2$ plane produces a closed curve which is a geodesic for the surface. The intuition of the curve being a geodesic comes from the work of [7], where they proved that the solution of energy functional of the Snake model is a geodesic on a Riemannian manifold equipped with a well-defined metric tensor. In this work we treat the neural network as a function approximator which learns to construct the three-dimensional regular surface. The network is guided to construct the regular surface such that its intersection with $\mathbb{R}^2$ plane is a geodesic and matches the boundary of the target object. For producing better results, we also incorporate *Deep Guided Filters* [15] to the network and train the network in an end-to-end manner. However, in our experiments we evaluate the proposed model with and without the deep guided filter to have thorough analysis of the performance of the model. The contributions of this paper can be summarized as follows,

- Introduction of a novel and geometric cost function for training convolutional neural networks for the salient object detection task.
- Integrating *Deep Guided Filters* into neural networks to produce better results.
- Introduction of an encoder-decoder network architecture for the mentioned task.

- *M. Matinkia is with the Department of Computer Engineering, Amirkabir University of Technology, Tehran, Iran.*

  E-mail: matinkia@aut.ac.ir
- *R. Safabakhsh is with the Department of Computer Engineering, Amirkabir University of Technology, Tehran, Iran.*

  E-mail: safa@aut.ac.ir

In Section 2, we review the related work on salient object detection and the methods which has combined the Level Set method and deep neural networks. In Section 3, we elaborate the proposed method as *Geodesically Driven Deep Level Sets* (GDDLS). In Section 4, we thoroughly examine our method on various datasets and show that our model achieves the state-of-the-art performance in salient object detection. Furthermore, we evaluate the performance of the variants of the GDDLS model. Finally, in Section 5, we provide a conclusion for this paper.

2 Related Works

In computer vision, object detection is performed by extracting informative features from the image. Information regarding the edges and boundaries in an image constitutes one of such features. *Active Contour Models* are powerful tools for extracting the boundaries of the objects in an image. These models can be categorized as *Snakes* and *Level Set* methods [8]. Snakes are Active Contour models which try to find the boundaries by minimizing an energy functional. In these models, a dynamic curve is evolved in directions which reduces the energy functional. This functional is designed such that its minimum occurs when the curve meets the boundary of the object. On the other hand, Level Set methods model the boundary of the objects as the intersection of a three-dimensional surface with $\mathbb{R}^2$ plane. The evolution *partial differential equations* of such model lead the surface to take a form such that the intersection matches the boundaries of the objects. A great advantage of the Level Set methods against Snake models is their ability to capture deformations in the topology of the curve. The early methods proposed for active contour models only incorporated the edge information from the image. This made these models prone to noises in the image. However, various methods were proposed to enhance active contour models and mitigate their problems by considering the information derived from regions [16], [17], [18] and shapes [19], [20] or using variational methods [16], [21], [22].

Caselles et al [7] showed that a slightly modified energy functional for a Snake model is equivalent to finding a geodesic on a Riemannian manifold with a determined metric tensor. This leads to the fact that the solution of the energy functional associated with a Snake model corresponds to a geodesic curve on a Riemannian manifold. This attempt provided a geometric point of view for Snake models and Level Set methods.

As a powerful tool, deep neural networks have shown great advantages in extracting features from various types of data, including visual data. Deep features extracted by convolutional neural networks are more informative and effective in various computer vision tasks such as salient objects detection. With their human-competitive performance, deep neural networks have been the state-of-the-art methods in object detection [23], [24], [25], [26], [27], [28], [29], [30]. However, there are a variety of methods which combine the Active Contour models and deep neural networks. For instance, Abdelsamea et al [31] propose a supervised ACM which combines variational Level Sets with the weights of the neurons of two Self-Organizing Maps. Rupprecht et al [32] propose Deep Active Contours and by using a convolutional neural network they learn a predictive vector field which in turn is used to guide the Snake model toward the region of interest. Hu et al [33] and Chen et al [34] formulate the Level Set method as the cost function of convolutional neural networks and they also incorporate the region information for better performance. Marcos et al [35] propose *Deep Structured Active Contours* in which they learn the parameterization of the ACM using convolutional neural networks.

3 Proposed Method

As mentioned in previous section, Caselles et al [7] demonstrated that solving a slightly modified energy functional for Active Contour models is equivalent to finding a closed geodesic in a Riemannian manifold with a specific metric structure. This metric structure is derived according to the Hamiltonian mechanics principles. In this work, we assume that the solution of the energy functional of the Active Contour model is a geodesic on a three-dimensional regular surface. Based on the principles of the regular surfaces theory and geodesics, we develop a local criterion for a planar curve to be a geodesic on a three-dimensional regular surface. Then, we try to construct the desired regular surface using a convolutional neural network. In other words, we develop a novel cost function to make a convolutional network learn to construct specific regular surfaces for which the boundaries of the foreground objects are geodesics. Finally, we show that this new formulation is not only more interpretable, but also leads to better performance in salient objects detection.

3.1 Geodesics of a Regular Surface

Introducing the definitions and principles of differential geometry cannot be included in this paper. We assume that the reader is familiar with the necessary concepts. However, the keen reader might refer to [36] for a thorough study of the matter.

We assume that we have a regular surface of the form

$$\mathbf{r}(x, y) = (x, y, \phi(x, y)), \tag{1}$$

in which $\phi(x, y)$ would be implemented by a convolutional neural network. For a regular surface, we can find an induced metric, namely *First Fundamental Form*, which is calculated as

$$I = \begin{bmatrix} E & F \\ F & G \end{bmatrix} \tag{2}$$

$$E = \mathbf{r}_x.\mathbf{r}_x \ , \ F = \mathbf{r}_x.\mathbf{r}_y \ , \ G = \mathbf{r}_y.\mathbf{r}_y, \tag{3}$$

where the subscripts indicate the derivatives with respect to the corresponding subscript and the "." represents the Euclidean inner product. For a regular surface of form (1), we can simply calculate the first fundamental form as

$$\begin{bmatrix} 1 + \phi_x^2 & \phi_x \phi_y \\ \phi_x \phi_y & 1 + \phi_y^2 \end{bmatrix}. \tag{4}$$

Suppose we have a parameterized curve in $\mathbb{R}^2$. Thus, we can represent this curve as

$$\gamma : [a, b] \to \mathbb{R}^2. \tag{5}$$

Note that a regular surface takes a point in $\mathbb{R}^2$ and produces a point on the regular surface S in $\mathbb{R}^3$. Since a regular surface is expressed as a diffeomorphic map, we can say that a parameterized curve on the surface S has a unique projection on the real plane $\mathbb{R}^2$. Having the projection of a curve on a regular surface S, and given the first fundamental form I of the surface, one can calculate the length of the parameterized curve $\gamma(t)$ on the surface S as

$$L_a^b[\gamma] = \int_a^b \sqrt{E\dot{x}^2 + 2F\dot{x}\dot{y} + G\dot{y}^2}. \tag{6}$$

A *geodesic* curve would arise by minimizing (6) with respect to the curve γ. In other words, the curve γ that minimizes the functional (6) is called a geodesic:

$$\min_{\gamma} L_a^b[\gamma] = \min_{\gamma} \int_a^b \sqrt{E\dot{x}^2 + 2F\dot{x}\dot{y} + G\dot{y}^2}. \tag{7}$$

Finding the Euler-Lagrange equations of the calculus of variation problem of (7), results in the *geodesic equations* expressed as:

$$\begin{aligned} \ddot{x} + \Gamma_{11}^1(\dot{x})^2 + \Gamma_{12}^1(\dot{x})(\dot{y})\Gamma_{22}^1(\dot{y})^2 = 0 \\ \ddot{y} + \Gamma_{11}^2(\dot{x})^2 + \Gamma_{12}^2(\dot{x})(\dot{y})\Gamma_{22}^2(\dot{y})^2 = 0, \end{aligned} \tag{8}$$

where Γ_{ij}^k indicate the *Christoffel symbols*, which are simply computed from only the first fundamental form by the formulas:

$$\begin{aligned} \Gamma_{11}^1 &= \frac{GE_u - 2FF_u + FE_v}{2(EG - F^2)} & \Gamma_{11}^2 &= \frac{2EF_u - EE_v - FE_u}{2(EG - F^2)} \\ \Gamma_{12}^1 &= \frac{GE_v - FG_u}{2(EG - F^2)} & \Gamma_{12}^2 &= \frac{EG_u - FE_v}{2(EG - F^2)} \\ \Gamma_{22}^1 &= \frac{2GF_v - GG_u - FG_v}{2(EG - F^2)} & \Gamma_{22}^2 &= \frac{EG_v - 2FF_v + FG_u}{2(EG - F^2)}. \end{aligned} \tag{9}$$

For a regular surface of form (1), the geodesic equations of (8) would be calculated as:

$$\begin{aligned} \ddot{x} + \frac{\phi_x\phi_{xx}}{1+\phi_x^2+\phi_y^2}(\dot{x})^2 + \frac{\phi_x\phi_{xy}}{1+\phi_x^2+\phi_y^2}(\dot{x})(\dot{y}) \\ + \frac{\phi_x\phi_{yy}}{1+\phi_x^2+\phi_y^2}(\dot{y})^2 = 0 \\ \ddot{y} + \frac{\phi_y\phi_{xx}}{1+\phi_x^2+\phi_y^2}(\dot{x})^2 + \frac{\phi_y\phi_{xy}}{1+\phi_x^2+\phi_y^2}(\dot{x})(\dot{y}) \\ + \frac{\phi_y\phi_{yy}}{1+\phi_x^2+\phi_y^2}(\dot{y})^2 = 0. \end{aligned} \tag{10}$$

Knowing the regular surface, one can simply calculate the geodesics on the surface by solving the system of ordinary differential equations of (10). However, the problem we handle in this work is the inverse of the problem of finding a geodesic for a regular surface. More precisely, our problem is defined as knowing the geodesics of a regular surface of the form (1), and knowing that this geodesic lies entirely on the $\mathbb{R}^2$ plane, how can we determine the regular surface of form (1). In the next section, we propose our method for finding such a surface and utilize it in salient objects detection.

3.2 Geodesically Driven Deep Level Sets

In the last section, we developed the system of equations of (10) for finding the geodesics of a regular surface of form (1). However, in the definition of our problem we assume that the boundary of an object is a geodesic on a regular surface. Therefore, we know the geodesics and we aim to determine the regular surface. Solving the system of equations of (10) for $\phi(x, y)$ is hard, and due to the lack of boundary and initial conditions, there is no guarantee to find a solution. However, using (10) we find a local criterion for a surface for which, a particular simple closed curve in $\mathbb{R}^2$ is a geodesic. Assuming that $\phi_y \neq 0$, by multiplying the second equation of (10) by $-\frac{\phi_x}{\phi_y}$ and adding the two equations we get

$$\phi_y\ddot{x} - \phi_x\ddot{y} = 0 \tag{11}$$

A careful look at (11) reveals that this equation is a vector product of the form

$$\begin{bmatrix} \ddot{x} \\ \ddot{y} \\ 0 \end{bmatrix} \times \begin{bmatrix} \phi_x \\ \phi_y \\ 0 \end{bmatrix} = \begin{vmatrix} x & y & z \\ \ddot{x} & \ddot{y} & 0 \\ \phi_x & \phi_y & 0 \end{vmatrix} = (0, 0, \phi_y\ddot{x} - \phi_x\ddot{y}). \tag{12}$$

Thus (11) being zero means that the outer product of (12) is zero, which subsequently means that the two vectors $[\ddot{x}, \ddot{y}, 0]$ and $[\phi_x, \phi_y, 0]$ are aligned (the angle between these two vectors is $k\pi$ for any integer k). If the curve lies on the $\mathbb{R}^2$ plane, the first vector shows the second derivative of the curve and represents the normal vector of the curve. In order to go further, first we need to define one more property of regular surfaces. For a regular surface $\mathbf{r}(x, y)$, the *unit normal vector* on each point of the surface is define as

$$\vec{\mathcal{N}} = \frac{\mathbf{r}_x \times \mathbf{r}_y}{\|\mathbf{r}_x \times \mathbf{r}_y\|}, \tag{13}$$

where $\times$ indicates the outer product. For a regular surface of form (1), the unit normal vector is computed as

$$\vec{\mathcal{N}} = \left(\frac{-\phi_x}{\sqrt{1+\phi_x^2+\phi_y^2}}, \frac{-\phi_y}{\sqrt{1+\phi_x^2+\phi_y^2}}, \frac{1}{\sqrt{1+\phi_x^2+\phi_y^2}} \right). \tag{14}$$

We claim that if the values ϕ_x^2 and ϕ_y^2 tend to infinity at points on the curve, the vector $[\phi_x, \phi_y, 0]$ being aligned with $[\ddot{x}, \ddot{y}, 0]$ is equivalent to the normal vector of (14) being aligned to $[\ddot{x}, \ddot{y}, 0]$. If ϕ_x^2 and ϕ_y^2 tend to infinity, the third component of the normal vector would be zero. Now, suppose that $[\phi_x, \phi_y, 0]$ and $[\ddot{x}, \ddot{y}, 0]$ are aligned. If we calculate the outer product of the vectors $[\ddot{x}, \ddot{y}, 0]$ and $\mathcal{N}$, we get

$$\begin{vmatrix} x & y & z \\ \ddot{x} & \ddot{y} & 0 \\ \frac{-\phi_x}{\sqrt{1+\phi_x^2+\phi_y^2}} & \frac{-\phi_y}{\sqrt{1+\phi_x^2+\phi_y^2}} & \frac{1}{\sqrt{1+\phi_x^2+\phi_y^2}} \end{vmatrix} = \left(\frac{\ddot{y}}{\sqrt{1+\phi_x^2+\phi_y^2}}, \frac{-\ddot{x}}{\sqrt{1+\phi_x^2+\phi_y^2}}, \frac{\ddot{y}\phi_x - \ddot{x}\phi_y}{\sqrt{1+\phi_x^2+\phi_y^2}} \right). \tag{15}$$

The third component of (15) is zero based on (11). The first two components are also zero as ϕ_x^2 and ϕ_y^2 tend to infinity. Thus, the two vectors would be aligned. Hence, we can conclude that if on the points of the intersection curve, the surface has normal vectors aligned with the normal vectors of the curve, then condition (11) is satisfied, and the curve is a geodesic. This result can be geometrically summarized as follows: in order for a simple closed curve on $\mathbb{R}^2$ to be a geodesic of a regular surface of form (1), the surface needs to be perpendicular on the $\mathbb{R}^2$ plane at the points of the curve (Fig. 1).

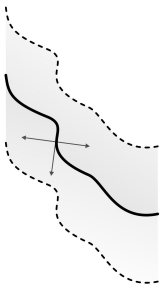


Fig. 1: The surface is perpendicular to the plane and the normal vectors are aligned

To recapitulate what we have developed so far, we have presented a local criterion for a curve to be a geodesic on a regular surface. Now, we will construct a cost function for a convolutional network to generate the desired surfaces. So far, we have concluded that for a curve to be a geodesic on a regular surface of form (1), the surface needs to be perpendicular on the plane on which the curve lies. In other words, the partial derivatives of the surface tend to infinity at the points of the curve (note that this does not contradict to the surface being regular [36]). This result can be summarized as

$$L_g(\phi) = \iint_\Omega \delta(x,y)(\phi_x^2 + \phi_y^2)dxdy, \tag{16}$$

where L_g represents the geodesics loss term, $\delta(.)$ is the Dirac delta function and Ω represents the domain of the image. However, this result specifies the surface only in a neighborhood of the curve. To develop further criteria for the surface, we use another concept in the theory of regular surfaces. The *Gaussian curvature* of a surface at each point on the surface is defined as

$$K = \frac{ef - g^2}{EF - G^2}, \tag{17}$$

where E, F, and G are the elements of the first fundamental form, and e, f, and g are elements of the *second fundamental form* defined by

$$\Pi = \begin{bmatrix} e & f \\ f & g \end{bmatrix} \tag{18}$$

$$e = \vec{\mathcal{N}}.\mathbf{r}_{xx}\ ;\ f = \vec{\mathcal{N}}.\mathbf{r}_{xy}\ ;\ g = \vec{\mathcal{N}}.\mathbf{r}_{yy}. \tag{19}$$

Gaussian curvature is an intrinsic geometric property of the surface depending only on the metric defined over the surface, and indicates a measure of the curvature of the surface. The Gaussian curvature for a regular surface of form (1) is calculated as

$$K(x,y) = \frac{\phi_{xx}\phi_{yy} - \phi_{xy}^2}{1 + \phi_x^2 + \phi_y^2} \tag{20}$$

We can use the Gaussian curvature of the surface for further specification of our surface. One reasonable choice which is in accordance with the theory of level sets in image processing is to choose a surface which has a positive Gaussian curvature on the points inside the curve and negative curvature on the points outside the curve. Thus, we can introduce another term for the loss function of the convolutional network which leads the network to construct surfaces with desirable curvature properties. This term is defined as

$$\begin{aligned} L_c(\phi) = &\iint_\Omega |H(\phi(x,y))K(x,y) - c_1|^2 dxdy \\ &+ \iint_\Omega |(1 - H(\phi(x,y)))K(x,y) - c_2|^2 dxdy. \end{aligned} \tag{21}$$

In (21), L_c represents the curvature loss and $H(.)$ is the Heaviside function. This loss drives the curvatures of the points inside and outside the curve into the constants c_1 and c_2. The last term required for the loss function of the convolutional network would be the one that matches the intersection of the surface and the $\mathbb{R}^2$ to the position of the curve. This term can be expressed as

$$L_s(\phi) = \iint_\Omega |H(\phi(x,y)) - gt(x,y)|^2 dxdy, \tag{22}$$

where $gt(x,y)$ is the ground truth image where each pixel is 1 if the pixel belongs to the foreground and the pixel is 0 if it belongs to the background. This loss function pushes the surface to a state where its intersection with xy plane results in the given curve. Putting equations (22), (16), and (21) together we can come up with a geometric loss function defined by

$$\begin{aligned} L(\phi) = &\iint_\Omega |H(\phi(x,y)) - gt(x,y)|^2 dxdy \\ &+ \lambda_1 \iint_\Omega \delta(x,y)(\phi_x^2 + \phi_y^2)dxdy \\ &+ \lambda_2 \iint_\Omega |H(\phi(x,y))K(x,y) - c_1|^2 dxdy \\ &+ \lambda_2 \iint_\Omega |(1 - H(\phi(x,y)))K(x,y) - c^2| dxdy, \end{aligned} \tag{23}$$

where λ_1 and λ_2 are regularization factors. The optimal values for the constants c_1 and c_2 can be calculated by setting the derivative of the (23) with respect to c_1 and c_2 to zero. Hence, by setting $\frac{\partial L(\phi)}{\partial c_1} = 0$ and $\frac{\partial L(\phi)}{\partial c_2} = 0$, we get the optimal values for the constants c_1 and c_2 as

$$\begin{aligned} c_1 &= \frac{\iint K(x,y)H(\phi(x,y))dxdy}{\iint dxdy} \\ c_2 &= \frac{\iint K(x,y)(1 - H(\phi(x,y)))dxdy}{\iint dxdy} \end{aligned} \tag{24}$$

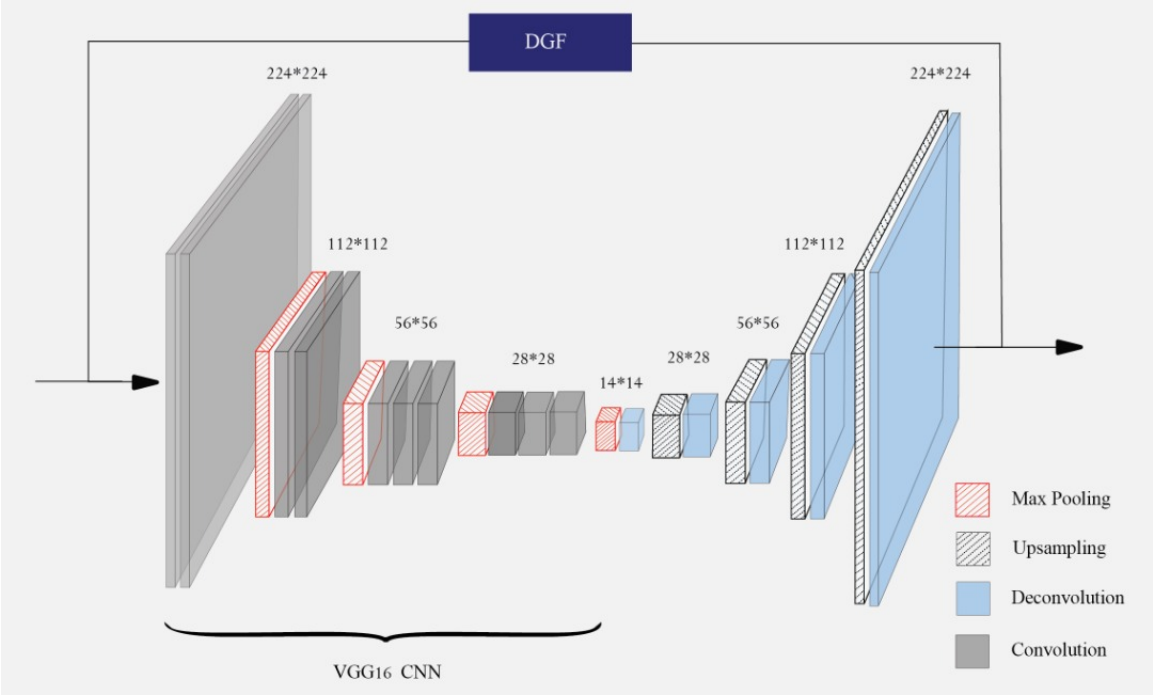


Fig. 2: The architecture of the proposed model.

3.3 Deep Guided Filters

Many recent methods in salient object detection have incorporated a form of filtering layer in their neural architecture to further enhance their output result ([33], [37]). Among these filtering methods, *Guided Filters* and their variants are of special interest. For instance, Hu et al [33], employ a *Guided Super-pixel Filtering* layer as the last layer of their network. Guided filters, first introduced by He et al [38], is a type of edge-preserving explicit image filter, which take a guidance image along with the input image to produce the output filtered image. Given a guidance image I and an input image p, guided filtering produces an output image q as a weighted average process:

$$q_i = \sum_j W_{ij}(I)p_j, \tag{25}$$

where i, j are pixel indices, and the weights W_{ij} are defined as

$$W_{ij} = \frac{1}{|\omega|^2} \sum_{k:(i,j)\in\omega_k} (1 + \frac{(I_i - \mu_k)(I_j - \mu_k)}{\sigma_k^2 + \epsilon}), \tag{26}$$

where ω_k is a window centered at pixel k with the mean μ_k and the variance σ_k^2. In this paper we use a more recent version of guided filters known as *Deep Guided Filters* introduced by Wu et al [15], which is designed for efficiently generating high resolution output given the corresponding low resolution input and high resolution guidance map. Here, the input to the guided filter is the output of our proposed model and the guidance map is the input image. Deep guided filter is added to the model as the last layer. Figure 2 illustrates the architecture of the model with and without the deep guided filter layer. The input image and the guidance map of the guided filter must have the same number of channels. Since the guidance map (the input image) has three channels and the input image (output of the model) has only one channel, we use a two-layer convolutional network to first transform input to 64 channels and then transform it back to 1 channel. This makes the deep guided filter even more powerful in the sense of edge preservation. Finally, the whole architecture can be trained end-to-end to produce better results.

In the next section we experiment with the proposed model with and without the deep guided filter layer and evaluate the performance of the proposed loss function.

4 Experimental Settings

The experiments are performed in six different configurations. These experiments are devised depending on whether the model has deep guided filter (DGF), and whether it is completely trained using (23) or it has been partially trained with *Binary Cross-Entropy*(BCE) and then fine-tuned using (23). However, the ablation study and hyperparameter tuning are performed on the configuration which led to the best results.

4.1 Implementation Details

To tackle the problem of differentiability of the Heaviside and Dirac delta functions, we employ the Approximated Heaviside Function proposed by Chan and Vese [39] which is

$$\begin{aligned} H_\epsilon(z) &= \frac{1}{2}(1 + \frac{2}{\pi}\arctan(\frac{z}{\epsilon})) \\ \delta_\epsilon(z) &= \frac{\partial H_\epsilon(z)}{\partial z} = \frac{1}{\pi}.\frac{\epsilon}{\epsilon^2 + z^2}. \end{aligned} \tag{27}$$

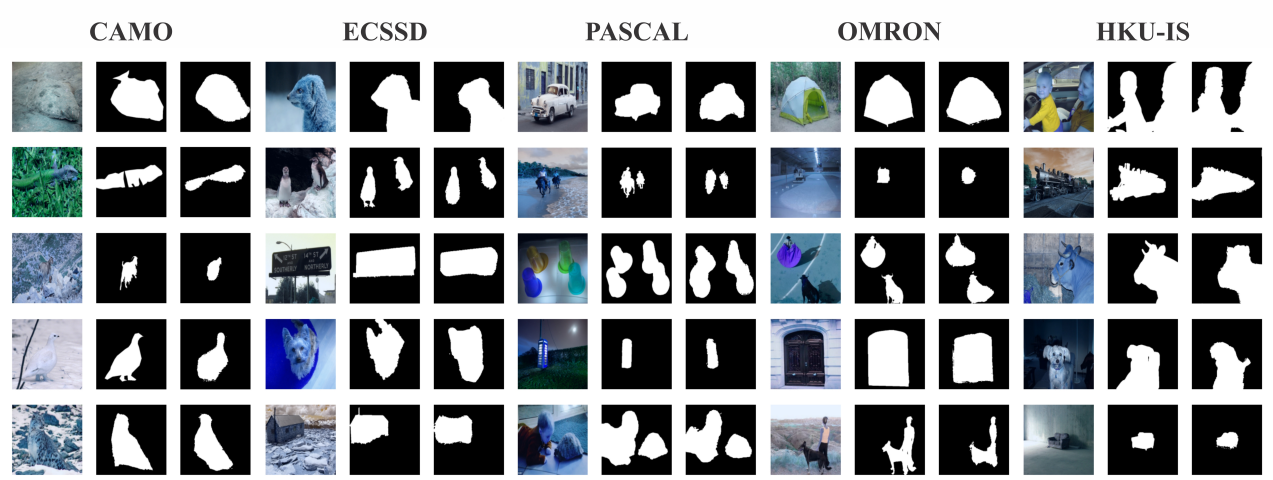


Fig. 3: The visual performance of our proposed model. For each dataset the first column represent the actual image, the second column represents the ground truth and the third column represents the output of our model.

TABLE 1: Hyperparameters settings - ϵ is the hyperparameter associated with Heaviside and Dirac functions and R is the radius of the guided filter

Name	Value
λ_1	0.3
λ_2	0.1
ϵ	$1/32$
R	8
Learning Rate	0.0001
Epochs	20

Furthermore, similar to [33], we use $\epsilon = \frac{1}{32}$ to calculate (27), although in subsection 4.5 we explore the effect of the different values of ϵ on the performance of the proposed model.
As Figure 2 illustrates, the final model is comprised of a pre-trained VGG16 network followed by a decoder network. The decoder contains six deconvolution and five upsampling layers. We have used Rectified Linear Units (ReLU) activation functions for hidden layers of the network. Since we use a pre-trained VGG16 network, the input images need to be rescaled to the size 224×224 and be normalized such that they have the mean $(0.485, 0.456, 0.406)$ and the variance $(0.229, 0.224, 0.225)$.
To train the model we use the Adam optimizer with the learning rate of 0.0001, and we set the values of the hyperparameters λ_1 and λ_2 to 0.3 and 0.1, respectively (in subsection 4.5 we provide a complete analysis on choosing the values of these hyperparameters). Finally, the model is implemented and evaluated in python and using the pyTorch package [40]. A summary of the used hyperparameters are shown in Table 1.

Regardless of the model having the DGF or not, a set of experiments are conducted by first training the model for 20 epochs using the *Binary Cross-Entropy* and then fine tuning the model using the proposed loss function for another 20 epochs.

4.2 Datasets

We have evaluated our model using five different datasets, namely **ECSSD** [41], **PASCAL** [42], **HKU-IS** [43], **OMRON** [44], and **CAMO** [45]. The **ECSSD** dataset contains 1000 images with structurally complex content. The **PASCAL-V12** dataset contains 2913 images with complex scenes. The ground truth images in this dataset are not binary, so we threshold them at 0.5 to have ground truths of salient objects. The **HKU-IS** dataset consists of 4447 challenging images with multiple objects, objects on the boundary and objects of low contrast. The **OMRON** dataset includes 5168 images with complex backgrounds. Finally, the **CAMO** dataset contains 1250 images of highly camouflaged objects. For each dataset we designate 80% of the data for training and the remaining data for testing the model.

4.3 Evaluation Metrics

In order to quantitatively compare our results with the existing works, we have used three different measures including Mean Absolute Error, Adaptive-F_β and ω-F_β. Here, we intend to briefly introduce the mentioned measures. The Mean Absolute Error is average per-pixel difference between the output saliency map and the ground truth. This metric is defined as

$$MAE = \frac{1}{W \times H} \sum_{x,y} |H(\phi(x,y)) - gt(x,y)|, \qquad (28)$$

where $gt(x,y)$ is the ground truth image and W and H are the width and the height of the image, respectively. Suppose that the *Precision* is defined as $Precision = \frac{|M \cap G|}{|M|}$ and $Recall = \frac{|M \cap G|}{|G|}$, where M is the output binary mask and G is the ground truth.Hence, the Adaptive-F_β is defined as

$$F_\beta = \frac{(1+\beta^2) \times Precision \times Recall}{\beta^2 \times Precision + Recall}, \qquad (29)$$

TABLE 2: Comparing the performance of different configuration of the proposed method. GDDLS represent the model without DGF and trained with (23). GDDLS + BCE indicates the model without DGF but pretrained using BCE and fine tuned using (23). BCE indicates the model without DGF and only trained by BCE. Configurations with the name DGF in them represent an architecture with deep guided filter. The best performance is marked with red.

Configuration	Dataset									
	CAMO		ECSSD		PASCAL		OMRON		HKU-IS	
	$MAE \Downarrow$	$F_\beta \Uparrow$	$MAE \Downarrow$	$F_\beta \Uparrow$	$MAE \Downarrow$	$F_\beta \Uparrow$	$MAE \Downarrow$	$F_\beta \Uparrow$	$MAE \Downarrow$	$F_\beta \Uparrow$
GDDLS	0.16	0.55	0.11	0.74	0.15	0.69	0.11	0.58	0.12	0.59
GDDLS + BCE	0.14	0.60	0.08	0.82	0.13	0.75	0.07	0.70	0.07	0.81
GDDLS + DGF	0.15	0.57	0.10	0.76	0.14	0.70	0.08	0.65	0.08	0.75
GDDLS + BCE + DGF	**0.13**	**0.61**	**0.08**	**0.83**	**0.13**	**0.76**	**0.07**	**0.71**	**0.06**	**0.83**
BCE	0.15	0.58	0.10	0.77	0.14	0.72	0.08	0.67	0.08	0.76
BCE + DGF	0.14	0.57	0.10	0.77	0.14	0.70	0.08	0.65	0.08	0.74

TABLE 3: Comparing the performance of GDDLS + BCE + DGF with other related methods. ANet is proposed for camouflaged dataset. Hence, we present its result only for the CAMO dataset

Model	Dataset									
	CAMO		ECSSD		PASCAL		OMRON		HKU-IS	
	$MAE \Downarrow$	$F_\beta \Uparrow$	$MAE \Downarrow$	$F_\beta \Uparrow$	$MAE \Downarrow$	$F_\beta \Uparrow$	$MAE \Downarrow$	$F_\beta \Uparrow$	$MAE \Downarrow$	$F_\beta \Uparrow$
DLS [33]	-	-	0.09	0.76	0.13	0.65	0.09	0.59	0.07	0.74
MTDS [46]	-	-	0.12	0.66	0.17	0.53	0.12	0.48	0.08	0.71
MDF [47]	-	-	0.11	0.69	0.15	0.58	0.09	0.55	0.13	0.56
MCDL [48]	-	-	0.11	0.67	0.15	0.57	0.09	0.54	0.10	0.63
ELD [49]	-	-	0.09	0.75	0.13	0.65	0.09	0.58	0.08	0.71
LEGS [50]	-	-	0.12	0.68	0.16	0.59	0.13	0.52	0.12	0.60
ANet [45]	0.13	**0.62**	-	-	-	-	-	-	-	-
GDDLS + BCE + DGF	**0.13**	**0.61**	**0.08**	**0.83**	**0.13**	**0.76**	**0.07**	**0.71**	**0.06**	**0.83**

where β^2 is typically set to 0.3 [33]. The ω-F_β measure which tries to overcome the flaws of the Adaptive-F_β measure [33], [51], is defined as

$$\omega\text{-}F_\beta = \frac{(1+\beta^2) \times Precision^\omega \times Recall^\omega}{\beta^2 \times Precision^\omega + Recall^\omega} \tag{30}$$

where $Precision^\omega$ and $Recall^\omega$ are calculated as explained in [51].

4.4 Performance Comparison

For the first four datasets **ECSSD**, **PASCAL**, **HKU-IS**, and **OMRON**, we compare our results with the results of deep learning methods such as DLS [33], MTDS [46], MDF [47], MCDL [48], ELD [49], and LEGS [50]. For the **CAMO** dataset we compare our method with Anabranch model [45] (who also provided the dataset). For a fair comparison we use the results reported by the corresponding authors.
Furthermore, we provide a separate comparison of different configurations of the proposed model. Table 2 shows the results of different configurations of the proposed method. As the table shows, the best performance is achieved by the configuration GDDLS + BCE + DGF which is shown in red.

Considering that the best performance is achieved using the BCE pre-trained model equipped with DGF which is then fine-tuned by (23), we compare the performance of the proposed method with the recent state-of-the-art methods mentioned before. Table 3 demonstrates this comparison. Figure 4, illustrates the Adaptive F_β for the datasets **ECSSD**, **PASCAL**, and **OMRON**.

TABLE 4: Various values of λ_1 and λ_2

Parameter	Values				
λ_1	0.001	0.01	0.3	1	5
λ_2	0.0001	0.001	0.01	0.1	1

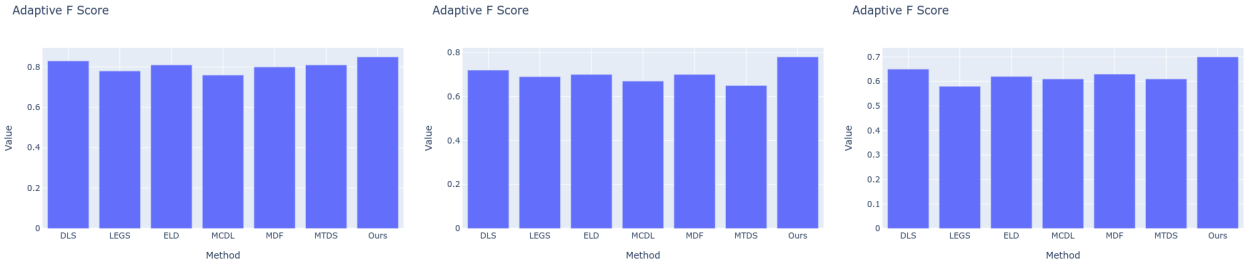
TABLE 5: Various values of ϵ

Parameter	Values				
ϵ	$\frac{1}{128}$	$\frac{1}{64}$	$\frac{1}{32}$	$\frac{1}{4}$	1

4.5 Additional Analysis

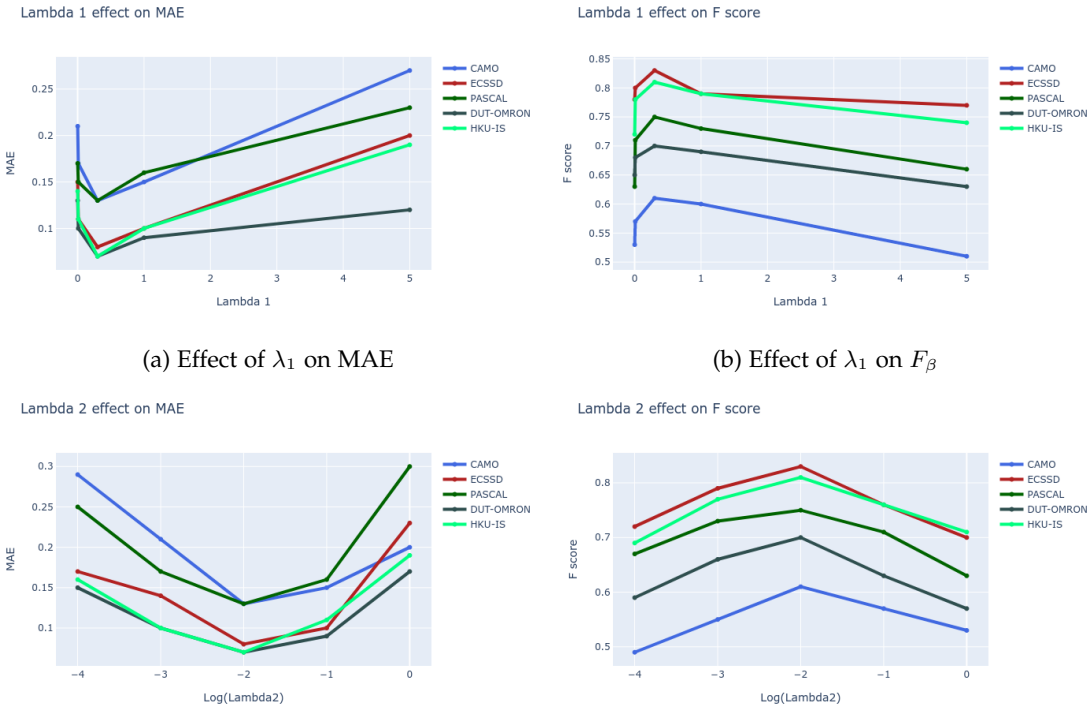
Here we provide the experiments leading to choosing the hyperparameters of the model. The important hyperparameters of the model are λ_1, λ_2, ϵ in the heaviside function approximation, and R which determines the radius of the guided filter. We begin by examining the effect of λ_1 and λ_2 on MAE and F_β. We select the proper values of λ_1 and λ_2 by sweeping the values demonstrated in Table 4. Finding the best values for λ_1 and λ_2 (values which yield the minimum MAE and maximum F_β), we fix one value and iterate through the other to plot the effect of the hyperparameter on the values of MAE and F_β. The results are presented in Figure 5. Using this information we came to the conclusion that the optimal values for λ_1 and λ_2 are 0.3 and 0.1, respectively.

Similar to [33], we iterate the values of ϵ from 1 to $\frac{1}{128}$ and track the effect of ϵ on MAE and F_β. Table 5 shows the various tested values for ϵ and Figure 6 illustrates the effect of ϵ on MAE and F_β in all the datasets.



(a) ECSSD (b) PASCAL (c) OMRON

Fig. 4: Adaptive F_β comparison between the datasets ECSSD, PASCAl and OMRON



(a) Effect of λ_1 on MAE (b) Effect of λ_1 on F_β

(c) Effect of $\log \lambda_2$ on MAE (d) Effect of $\log \lambda_2$ on F_β

Fig. 5: The effect of λ_1 and λ_2 on MAE and F_β score.

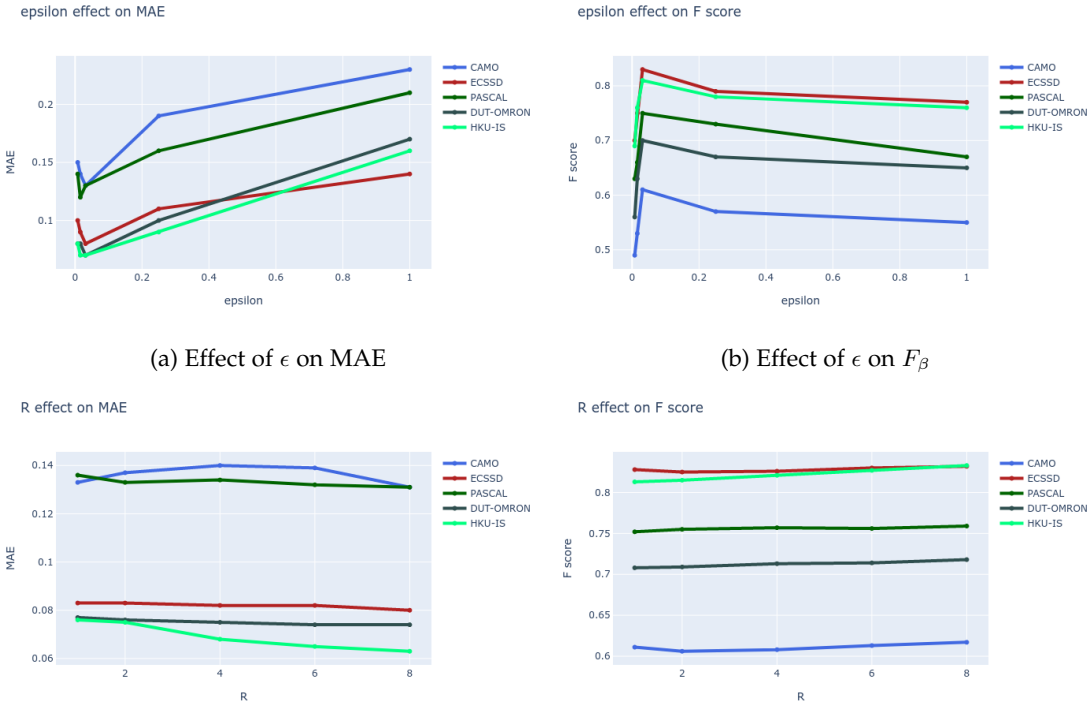
TABLE 6: Various values of R

Parameter	Values				
R	1	2	4	6	8

Based on the results shown in Figures 6a and 6b, we set the value of ϵ to $\frac{1}{32}$. Finally, we investigate the effect of the radius of the guided filter on the performance of the model and choose the value which lead to minimum MAE and maximum F_β. Once again we fix the architecture as GDDLS + BCE + DGF, and use the best hyperparameters found so far, and we iterate through the R values shown in Table 6. Figures 6c and 6d illustrate the effect of R on the values of MAE and F_β. Based on these experiments we set the value of R to 8.

5 Conclusion

In this paper we presented a novel cost function for convolutional neural networks supported by a geometric intuition and concept. Based on regular surfaces theory, we developed local criteria for a surface by which, its intersection with the $\mathbb{R}^2$ plane is a geodesic. Similarly, based on the Level Set method, we assumed that the intersection curve is the boundary of the salient object in the image. Combining these concepts, we introduced a cost function for training convolutional neural networks for the salient objects detection task. Furthermore, we incorporated deep guided filters into our proposed model to produce better results. The final architecture is trained in an end-to-end manner. This work mainly demonstrates the significance of merging geometric concepts into the deep learning paradigm.



(a) Effect of ϵ on MAE

(b) Effect of ϵ on F_β

(c) Effect of R on MAE

(d) Effect of R on F_β

Fig. 6: The effect of different values of ϵ and R on MAE and F_β score

REFERENCES

[1] Bharath Hariharan, Pablo Arbeláez, Ross Girshick, and Jitendra Malik. Simultaneous detection and segmentation. In *European Conference on Computer Vision*, pages 297–312. Springer, 2014.

[2] Bharath Hariharan, Pablo Arbeláez, Ross Girshick, and Jitendra Malik. Hypercolumns for object segmentation and fine-grained localization. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 447–456, 2015.

[3] Jifeng Dai, Kaiming He, and Jian Sun. Instance-aware semantic segmentation via multi-task network cascades. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3150–3158, 2016.

[4] Andrej Karpathy and Li Fei-Fei. Deep visual-semantic alignments for generating image descriptions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3128–3137, 2015.

[5] Michael Kass, Andrew Witkin, and Demetri Terzopoulos. Snakes: Active contour models. *International journal of computer vision*, 1(4):321–331, 1988.

[6] Yuwei Wu, Yuanquan Wang, and Yunde Jia. Adaptive diffusion flow active contours for image segmentation. *Computer Vision and Image Understanding*, 117(10):1421–1435, 2013.

[7] Vicent Caselles, Ron Kimmel, and Guillermo Sapiro. Geodesic active contours. *International journal of computer vision*, 22(1):61–79, 1997.

[8] Rafael C Gonzalez, Richard Eugene Woods, and Steven L Eddins. *Digital image processing using MATLAB*. Pearson Education India, 2004.

[9] Pedro Felzenszwalb, David McAllester, and Deva Ramanan. A discriminatively trained, multiscale, deformable part model. In *2008 IEEE conference on computer vision and pattern recognition*, pages 1–8. IEEE, 2008.

[10] Pedro F Felzenszwalb, Ross B Girshick, and David McAllester. Cascade object detection with deformable part models. In *2010 IEEE Computer society conference on computer vision and pattern recognition*, pages 2241–2248. IEEE, 2010.

[11] Pedro F Felzenszwalb, Ross B Girshick, David McAllester, and Deva Ramanan. Object detection with discriminatively trained part-based models. *IEEE transactions on pattern analysis and machine intelligence*, 32(9):1627–1645, 2009.

[12] Yichen Wei, Fang Wen, Wangjiang Zhu, and Jian Sun. Geodesic saliency using background priors. In *European conference on computer vision*, pages 29–42. Springer, 2012.

[13] Qiong Yan, Li Xu, Jianping Shi, and Jiaya Jia. Hierarchical saliency detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1155–1162, 2013.

[14] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[15] Huikai Wu, Shuai Zheng, Junge Zhang, and Kaiqi Huang. Fast end-to-end trainable guided filter. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1838–1847, 2018.

[16] Tony F Chan and Luminita A Vese. Active contours without edges. *IEEE Transactions on image processing*, 10(2):266–277, 2001.

[17] Shawn Lankton and Allen Tannenbaum. Localizing region-based active contours. *IEEE transactions on image processing*, 17(11):2029–2039, 2008.

[18] Sijie Niu, Qiang Chen, Luis De Sisternes, Zexuan Ji, Zeming Zhou, and Daniel L Rubin. Robust noise region-based active contour model via local similarity factor for image segmentation. *Pattern Recognition*, 61:104–119, 2017.

[19] Sahirzeeshan Ali and Anant Madabhushi. An integrated region-, boundary-, shape-based active contour for multiple object overlap resolution in histological imagery. *IEEE transactions on medical imaging*, 31(7):1448–1460, 2012.

[20] Daniel Cremers, Florian Tischhäuser, Joachim Weickert, and Christoph Schnörr. Diffusion snakes: Introducing statistical shape knowledge into the mumford-shah functional. *International journal of computer vision*, 50(3):295–313, 2002.

[21] David Bryant Mumford and Jayant Shah. Optimal approximations by piecewise smooth functions and associated variational problems. *Communications on pure and applied mathematics*, 1989.

[22] Chunming Li, Chenyang Xu, Changfeng Gui, and Martin D Fox. Level set evolution without re-initialization: a new variational formulation. In *2005 IEEE computer society conference on computer*

vision and pattern recognition (CVPR'05), volume 1, pages 430–436. IEEE, 2005.

[23] Nian Liu and Junwei Han. Dhsnet: Deep hierarchical saliency network for salient object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 678–686, 2016.

[24] Rui Zhao, Wanli Ouyang, Hongsheng Li, and Xiaogang Wang. Saliency detection by multi-context deep learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1265–1274, 2015.

[25] Jianming Zhang, Stan Sclaroff, Zhe Lin, Xiaohui Shen, Brian Price, and Radomir Mech. Unconstrained salient object detection via proposal subset optimization. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5733–5742, 2016.

[26] Qibin Hou, Ming-Ming Cheng, Xiaowei Hu, Ali Borji, Zhuowen Tu, and Philip HS Torr. Deeply supervised salient object detection with short connections. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3203–3212, 2017.

[27] Nian Liu, Junwei Han, and Ming-Hsuan Yang. Picanet: Learning pixel-wise contextual attention for saliency detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3089–3098, 2018.

[28] Yi Liu, Qiang Zhang, Dingwen Zhang, and Jungong Han. Employing deep part-object relationships for salient object detection. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1232–1241, 2019.

[29] Qi Qi, Sanyuan Zhao, Jianbing Shen, and Kin-Man Lam. Multi-scale capsule attention-based salient object detection with multi-crossed layer connections. In *2019 IEEE International Conference on Multimedia and Expo (ICME)*, pages 1762–1767. IEEE, 2019.

[30] Xuebin Qin, Zichen Zhang, Chenyang Huang, Chao Gao, Masood Dehghan, and Martin Jagersand. Basnet: Boundary-aware salient object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7479–7489, 2019.

[31] Mohammed M Abdelsamea, Giorgio Gnecco, and Mohamed Medhat Gaber. An efficient self-organizing active contour model for image segmentation. *Neurocomputing*, 149:820–835, 2015.

[32] Christian Rupprecht, Elizabeth Huaroc, Maximilian Baust, and Nassir Navab. Deep active contours. *arXiv preprint arXiv:1607.05074*, 2016.

[33] Ping Hu, Bing Shuai, Jun Liu, and Gang Wang. Deep level sets for salient object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2300–2309, 2017.

[34] Xu Chen, Bryan M Williams, Srinivasa R Vallabhaneni, Gabriela Czanner, Rachel Williams, and Yalin Zheng. Learning active contour models for medical image segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 11632–11640, 2019.

[35] Diego Marcos, Devis Tuia, Benjamin Kellenberger, Lisa Zhang, Min Bai, Renjie Liao, and Raquel Urtasun. Learning deep structured active contours end-to-end. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8877–8885, 2018.

[36] Manfredo P Do Carmo. *Differential geometry of curves and surfaces: revised and updated second edition*. Courier Dover Publications, 2016.

[37] Qiong Wang, Lu Zhang, Wenbin Zou, and Kidiyo Kpalma. Salient video object detection using a virtual border and guided filter. *Pattern Recognition*, 97:106998, 2020.

[38] Kaiming He, Jian Sun, and Xiaoou Tang. Guided image filtering. In *European conference on computer vision*, pages 1–14. Springer, 2010.

[39] Tony F Chan and Luminita A Vese. Active contours without edges. *IEEE Transactions on image processing*, 10(2):266–277, 2001.

[40] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. 2017.

[41] Qiong Yan, Li Xu, Jianping Shi, and Jiaya Jia. Hierarchical saliency detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1155–1162, 2013.

[42] Sara Vicente, Joao Carreira, Lourdes Agapito, and Jorge Batista. Reconstructing pascal voc. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 41–48, 2014.

[43] Guanbin Li and Yizhou Yu. Visual saliency based on multiscale deep features. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5455–5463, 2015.

[44] Chuan Yang, Lihe Zhang, Huchuan Lu, Xiang Ruan, and Ming-Hsuan Yang. Saliency detection via graph-based manifold ranking. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3166–3173, 2013.

[45] Trung-Nghia Le, Tam V Nguyen, Zhongliang Nie, Minh-Triet Tran, and Akihiro Sugimoto. Anabranch network for camouflaged object segmentation. *Computer Vision and Image Understanding*, 184:45–56, 2019.

[46] Xi Li, Liming Zhao, Lina Wei, Ming-Hsuan Yang, Fei Wu, Yueting Zhuang, Haibin Ling, and Jingdong Wang. Deepsaliency: Multi-task deep neural network model for salient object detection. *IEEE transactions on image processing*, 25(8):3919–3930, 2016.

[47] Guanbin Li and Yizhou Yu. Visual saliency based on multiscale deep features. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5455–5463, 2015.

[48] Rui Zhao, Wanli Ouyang, Hongsheng Li, and Xiaogang Wang. Saliency detection by multi-context deep learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1265–1274, 2015.

[49] Gayoung Lee, Yu-Wing Tai, and Junmo Kim. Deep saliency with encoded low level distance map and high level features. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 660–668, 2016.

[50] Lijun Wang, Huchuan Lu, Xiang Ruan, and Ming-Hsuan Yang. Deep networks for saliency detection via local estimation and global search. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3183–3192, 2015.

[51] Ran Margolin, Lihi Zelnik-Manor, and Ayellet Tal. How to evaluate foreground maps? In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 248–255, 2014.

Mohammadjavad Matinkia was born in Tehran, Iran, in 1992. He received his B.S. degree in electrical engineering from University of Tehran, Iran in 2017, and the M.S. degree in computer engineering from Amirkabir University of Technology in 2020. His research interests include computer vision and deep learning. He has been a member of Deep Learning and Computer Vision Laboratories at Amirkabir University of Technology since 2017.

Reza Safabakhsh was born in Isfahan, Iran, in 1953. He received the B.S. degree in electrical engineering from Sharif University of Technology, Tehran, Iran, and the M.S. and Ph.D. degrees in electrical and computer engineering from the University of Tennessee, Knoxville, in 1980 and 1986, respectively. He worked at the Center of Excellence in Information Systems, Nashville, TN from 1986 to 1988. Since 1988, he has been with the Computer Engineering Department, Amirkabir University of Technology, Tehran, Iran, where he is currently a Professor and the director of the Deep Learning and Computer Vision Laboratories. His current research interests include deep learning, computer vision, and financial engineering. Dr. Safabakhsh is a member of the IEEE and several honor societies, including Phi Kappa Phi and Eta Kapa Nu. He was the founder and a member of the Board of Executives of the Computer Society of Iran, and was the President of this society for the first 4 years.