Improving FLIM-based Salient Object Detection Networks with Cellular Automata

Felipe Crispim R. Salvagnini*†, Jancarlo F. Gomes‡, Cid A. N. Santos†,
Silvio Jamil F. Guimarães§ and Alexandre X. Falcão*
*Institute of Computing, University of Campinas, Campinas, Brazil
†Computational Photography Department, Eldorado Institute, Campinas, Brazil
‡School of Medical Sciences, University of Campinas, Campinas, Brazil
§Institute of Computing Sciences, Pontifical Catholic University of Minas Gerais, Belo Horizonte, Brazil

Abstract—**Despite the success of Deep-Learning-based (DL) methods on Salient Object Detection (SOD), the need for abundantly labeled data and the high complexity of the network architectures limit their applications. Feature Learning from Image Markers (FLIM) is a recent methodology to build convolutional encoders with minimal human effort in data annotation. More recently, a FLIM encoder has been combined with an adaptive decoder to build flyweight FLIM networks for SOD, requiring only user-drawn markers in discriminative regions of a few (*e.g.,* 4) images to train the entire model with no backpropagation. Furthermore, due to the data scarcity in some applications, using Cellular Automata (CA) may help compute better saliency maps. However, the initialization of the CA could be a problem since it is based on user input, priors, or randomness. Here, we propose a new strategy for CA initialization via a FLIM-based SOD network. In summary, CA interprets pixels of an initial saliency map as cells and cleverly designs transition rules to generate an improved saliency map through the evolution and interaction of each cell and its neighbors using the original pixel properties. CA requires initializing the cell's states, where methods diverge. By exploring the saliency map of a FLIM network, we circumvent the CA initialization problem and improve FLIM saliencies. Experiments in two challenging medical datasets demonstrate improvements in FLIM-based SOD, with results comparable to two state-of-the-art DL methods fine-tuned under data scarcity.**

I. Introduction

When a neuroradiologist – or even an untrained person – examines a magnetic resonance image, abnormalities such as tumor lesions may be evident. Similarly, parasite eggs in a microscopy image stand out for a parasitologist. Salient Object Detection (SOD) methods can generate a saliency map in which such objects are brighter than the background.

SOD methods may be categorized into two approaches: top-down and bottom-up [1]. Top-down approaches focus on extracting high-level features through supervised learning, which requires annotated data [2]–[4]. Bottom-up approaches employ low-level features – local properties such as color and texture – and statistics to identify foreground regions [5].

Deep-learning-based SOD methods (DL-SOD) use an encoder to extract high-level features and, subsequently, a decoder to estimate a saliency object map upsampled to the input image size. Such methods usually employ a U-shaped Network [6]. Recent SOD models further explore multi-scale feature extractions to improve results. For instance, BasNet employs a deeply-supervised encoder-decoder and a hybrid boundary-aware loss to learn at pixel, patches, and map levels [3]. $\mathbf{U^2}$-Net, on the other hand, employs nested U-shaped architectures to improve high-level feature extraction at multiple resolutions (intra-stage multi-scale features) [2].

Regardless of deep-learning efficacy, they are data-hungry, and when faced with scarce data, overfitting is a recurrent scenario. Models learn to solve training data but do not learn discriminative features. In some areas, such as medicine, this problem is even worse. Manual labeling ground-truth data for medical image analysis tasks is time-consuming, error-prone, and labor-intensive [7]. Suitable for data scarcity, some works use Cellular Automata (CA) for medical and natural images, where a CA model is executed for each input [1], [8]–[11]. CA [9], proposed in 1951, interprets an image as a lattice of cells. Each pixel is a cell whose state (foreground/background) evolves from interactions with neighboring cells.

CA enables SOD methods using low-level features (image intensity) [11], [12] and high-level features [1], [10]. However, CA requires the state initialization of each cell. Methods commonly initialize states from user inputs, like a line drawn on image regions [11]–[13]. Others try random initialization [8] or priors (as background or contrast priors) [1]. None has taken advantage of high-level features during initialization; only user-based methods employ the expert's knowledge but require the user for every inference image.

Feature Learning from Image Markers [14], [15] (FLIM) is a recent methodology to build convolutional encoders with minimal human effort in data annotation. It addresses the data scarcity problem of DL-SOD methods since a FLIM encoder can be trained from user-drawn markers (scribbles, disks) on discriminative regions of a few (less than 10) images without backpropagation. Moreover, flyweight FLIM networks with a decoder that adapts to each input image can generate saliency object maps [16], making the entire network backpropagation-free. Hence, learning does not require ground truth, only weak annotation (*e.g.,* scribbles) on a few images, avoiding overfitting and enabling non-differentiable operations.

We propose a SOD method that combines a FLIM network and CA to provide a novel CA initialization, employing expert knowledge and high-level features. Our methodology is based on three different phases: (1) Design of a FLIM network; (2) CA initialization; and (3) CA evolution. A user inserts markers on a few selected images (design phase) and defines the FLIM-encoder architecture. Then, images are fed into the FLIM encoder for inference, where the extracted features are adaptively decoded into an intermediary saliency map for CA initialization. Finally, the CA evolves using the image properties to generate a final saliency map. Unlike other user-based CAs, **we do not require user interaction for every image**. To study the behavior of this new approach in real cases, we have applied it to two challenging medical datasets: Brain tumors (glioblastomas) and Parasite Eggs. (*Schistosoma Mansoni*). Performance on those datasets showcases our method in two different domains: gray-scale MRI images and RGB microscopy images.

For scenarios where few labeled training data is available, the main contributions of this work are threefold and may be summarized as follows:

1) We demonstrate that the FLIM-network-based initialization of a CA improves the performance of a FLIM network, with results comparable to two state-of-the-art DL-SOD models (fine-tuned with data scarcity);
2) We investigate evolving CA with low-level and high-level features using FLIM-encoder feature maps;
3) We evaluate CA initialization with DL-SOD methods, previously fine-tuned with data scarcity.

II. Theoretical Background

This section builds the intuition behind the FLIM methodology and introduces CA concepts, showing the main steps in its evolution. CA initialization methods require user intervention or ad-hoc criteria for each image, even on testing. We propose a novel method to initialize CA states by automating the process based on the saliency map $\mathcal{S}$ of a FLIM network.

A. Feature Learning from Image Markers (FLIM)

FLIM [14]–[16] consistently shows prominent results whenever annotated data are scarce. Deep-learning models often overfit without generalizing to unseen data when learning with little data. On the other hand, the FLIM methodology aims to exploit user knowledge in training image selection, discriminative region identification for filter estimation, filter evaluation, and filter selection, allowing the construction of a CNN encoder layer by layer. In this work, the user sets an encoder architecture, selects representative images, and identifies discriminative regions for the automatic filter estimation of all convolutional layers. Each encoder layer contains marker-based normalization, convolution with a filter bank, ReLU activation, and max-pooling. Except for marker-based normalization, the remaining operations are well-known. However, the definitions and geometrical interpretation below provide vital insights into FLIM.

Let $\mathbf{I} = (D_I, \vec{I}) \in \mathcal{D}$ be an image with m channels from a dataset $\mathcal{D}$, where $D_I \subset Z^2$ is the image domain and $\vec{I}(p) \in \mathbb{R}^m$ assigns m feature values to every pixel $p \in D_I$. The vectorization of the image features in a squared region centered at p, with size $k \times k \times m$, defines a patch vector $\vec{P}_p \in \mathbb{R}^{k\times k\times m}$. These definitions are valid for grayscale images ($m = 1$), color images ($m = 3$), and feature maps ($m \geq 1$) created by convolution with filter banks.

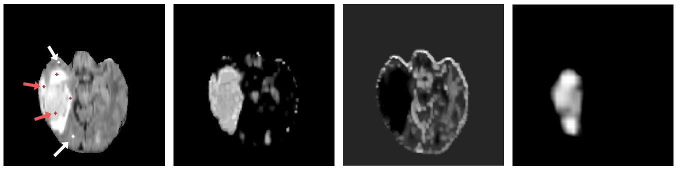

(a) (b) (c) (d)

Fig. 1. Adaptive decoder intuition: (a) input image and markers (red and white dots); (b) channel i with $w_i = 1$ (foreground); (c) channel j with $w_j = -1$ (background); (d) Resulting saliency map $\mathcal{S}$.

The user selects a subset $C_I \subseteq \mathcal{D}$ (usually, $|C_I| << |\mathcal{D}|$) with a few representative images (*e.g.,* 5) and draws markers (balls, scribbles) on discriminative regions of every image $\mathbf{I} \in C_I$, creating a set $\mathcal{M}_I$ of patch vectors $\vec{P}_p$ for pixels p marked in image $\mathbf{I}$. Knowledge of the target domain and an understanding of $\mathcal{D}$ is required for both user's actions, and these actions are needed only for the input images. Let $\mathcal{M} = \bigcup_{\mathbf{I}\in C_I} \mathcal{M}_I$, the patch vectors in $\mathcal{M}$ are normalized by z-score, and the normalization parameters are applied to all patch vectors from any image $\mathbf{I} \in \mathcal{D}$. This operation constitutes the *marker-based image normalization*. It essentially centralizes the dataset $\mathcal{M}$ of patch vectors and corrects distortions among the main axes of $\mathbb{R}^{k\times k\times m}$.

The user sets n filters per marker when defining the encoder architecture. A clustering of normalized patch vectors is computed for each marker by the k-means algorithm, generating n clusters per marker, and the center of each cluster defines the weight vector of a filter (kernel) $\vec{K}_i \in \mathbb{R}^{k\times k\times m}$. The union of the kernels from all markers defines the filter bank of the current layer. The convolution between a marker-based normalized image $\mathbf{I}$ and a kernel $\vec{K}_i$ creates an image $\mathbf{J}$ with pixel values $J_i(p) = \langle \vec{P}_p, \vec{K}_i \rangle$. Given a filter bank $\{\vec{K}_i\}_{i=1}^{n\times M}$, with $n \times M$ filters obtained from M markers, $\mathbf{J} = (D_J, \vec{J})$ will be a multichannel feature map with $n \times M$ channels and $\vec{J}(p) = (J_1(p), J_2(p), \ldots, J_{n\times M}(p)) \in \mathbb{R}^{n\times M}$.

A kernel $\vec{K}_i$ can be interpreted as a vector orthogonal to a hyperplane passing through the origin of $\mathbb{R}^{k\times k\times m}$. $\vec{P}_p$ is a point in $\mathbb{R}^{k\times k\times m}$ and $J_i(p)$ is the distance between the point and the hyperplane. Depending on which side of the hyperplane the point is, the distance is positive or negative. ReLU eliminates points on the negative side while max-pooling aggregates nearby activations. By forcing unit norm $\|\vec{K}_i\| = 1$ to all kernels, $J_i(p)$ is not amplified by the magnitude of $\vec{K}_i$. Marker-based normalization dismisses bias by centralizing the clusters around the origin of $\mathbb{R}^{k\times k\times m}$.

Patches extraction, marker-based normalization, and clustering operations repeat for the design of every encoder layer. They use feature maps of the previous layer (from images of $\mathcal{C}_I$) and the markers mapped to the feature maps. Then, the FLIM encoder can extract feature maps from any image in $\mathcal{D}$.

In [16], the authors introduce an essential characteristic of FLIM encoders. Markers drawn in the foreground and background (Figure 1a) generate feature maps where each channel activates for foreground or background regions (Figures 1b-1c). Decoding the feature map could be interpreted as a weighted average of the channels, followed by activation ϕ, which outputs a saliency map $\mathcal{S}$ (Figure 1d). Therefore, a decoder with positive weights to foreground channels and negative weights to background ones should create a salience map with fewer false positives, such that the object is the brightest component in the map. However, each feature-map channel weight may change according to the input image (negative or positive), requiring an adaptive decoder. The authors present such decoders for two datasets in [16]. We define a slight modification of an adaptive decoder from [17], suitable to our problem, which estimates a saliency map $\mathcal{S}$

$$\mathcal{S}(p) = \phi(\langle \vec{J}(p), \vec{w} \rangle), \tag{1}$$

where $\vec{w} = (w_1, w_2, \ldots, w_{n \times M})$, $\mathbf{J} = (D_J, \vec{J})$ is the feature map of the last encoder layer, and ϕ is ReLU. The image domain D_J is first interpolated to be D_I. Each $w_i \in \{-1, 0, 1\}$ assigns a weight to a feature-map channel J_i. Let μ_{J_i} be the mean activation of channel J_i, $\mathcal{T}$ be the Otsu threshold of the distribution $\{\mu_{J_i}\}_{i=1}^{n \times M}$, and σ^2 be the standard deviation of that distribution. The ratio between the number of pixels above the Otsu threshold of channel J_i and the feature map size $|D_J|$ is defined as a_i. Given A_1 and A_2 thresholds, w_i is

$$w_i = \begin{cases} +1, & \text{if} \quad \mu_{J_i} \leq \mathcal{T} - \sigma^2 \text{ and } a_i < A_1, \\ -1, & \text{if} \quad \mu_{J_i} \geq \mathcal{T} + \sigma^2 \text{ and } a_i > A_2, \\ 0, & \text{otherwise.} \end{cases} \tag{2}$$

B. Cellular Automata

Introduced in 1951 by John von Neumann, Cellular Automata (CA) is a discrete evolving model [9]. CA, formally defined as a triple $(\mathbb{S}, N, \delta)$, operates over a lattice of cells changing their states over time given a transition function. For images, CA requires a set D_I (image domain) of cells, a state description $\mathbb{S}$ for each cell $p \in D_I$, a definition of a cell neighborhood $N(p)$, and a local transition rule δ.

A cell state $\mathbb{S}$ at time t contains a strength $\theta_l^t(p)$ with label $l = L(p) \in \{0, 1\}$ (background, foreground). CA starts from an initial cell state $\mathbb{S}_p^0$ and evolves given a transition function $\delta : \mathbb{S}_p^t \to \mathbb{S}_p^{t+1}$, taking into account g and θ_l^t within $N(p)$, until it converges. According to [12], g may be defined by

$$g(p, q) = \begin{cases} e^{\beta||\vec{I}(p) - \vec{I}(q)||_2}, & \text{if } Y(p) > Y(q) \text{ \& } L(q) = 1 \\ e^{||\vec{I}(p) - \vec{I}(q)||_2}, & \text{otherwise,} \end{cases} \tag{3}$$

where Y is the luminance component from $\vec{I}$ for color images. For gray-scale images, $Y = I_1$. Parameter β smooths the weight reduction when transitioning from dark to bright areas.

In summary, **a cell q propagates its state to a cell p at time** $t+1$, if both cells are similar (close pixel colors) and if $\theta_l^t(q) > \theta_l^t(p)$, where $g(p, q) \times \theta_l^t(q) > \theta_l^t(p)$.

III. Salient Object Detection using FLIM Network and Cellular Automata

Our method is illustrated in Figure 2. The user interacts only with the design of the FLIM network. Given an input image **I**, the FLIM network outputs a saliency map $\mathcal{S}$. We normalize $\mathcal{S}$ whitin $[0, 1]$, and set cells' foreground strengths $\theta_1^0(p)$ using the normalized saliency values. The background strength $\theta_0^0(p)$ is initialized according to the task (see Experimental Setup). The label map L at $t = 0$ is initilized by setting $L(p) \leftarrow 1$, if $\theta_1^0(p) > 0$, and 0 otherwise.

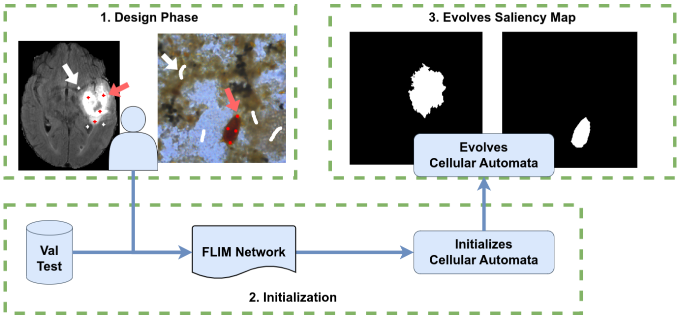


Fig. 2. Proposed method. (1) The user draws markers (red and white dots/scribbles) on selected images, generating a FLIM Network (FLIM Encoder + Adaptive Decoder); (2) At inference time, the FLIM Network initializes CA; (3) CA evolves the final saliency map.

Algorithm 1 details the cell state evolution for each label $l \in \{0, 1\}$ using a Moore Neighborhood (8-neighbors). It is inspired in [12] and is first executed for $l = 1$ and then for $l = 0$. Given an input state and label maps, θ_l^0 and L, *Line 2* initializes a variable *dist* used for convergence detection in *Line 3* of the main loop. In *Line 4*, a next-state map θ_l^{t+1} stores the current state map θ_l^t to keep unconquered cells. From *Lines 5* to *15*, θ_l^{t+1} evolves such that a cell p is conquered by a neighboring cell q if $g(p, q) \times \theta_l^t(q) > \theta_l^t(p)$, updating strength $\theta_l^{t+1}(p)$ and label $L(p)$ in *Lines 10* and *11*. We have used Eq. 3 for computing g. The convergence variable is updated in *Line 16*. Finally, the current-state map θ_l^t is updated as the next state in *Line 17*. *Line 18* increments the iteration t. The output states θ_l^t generates a object probability map O

$$O(p) = \frac{ln(\theta_0^t(p))}{ln(\theta_0^t(p)) + ln(\theta_1^t(p))} \tag{4}$$

The object is finally defined by pixels with $O(p) > 0.6$, set empirically from experimental results. Figure 3 illustrates the

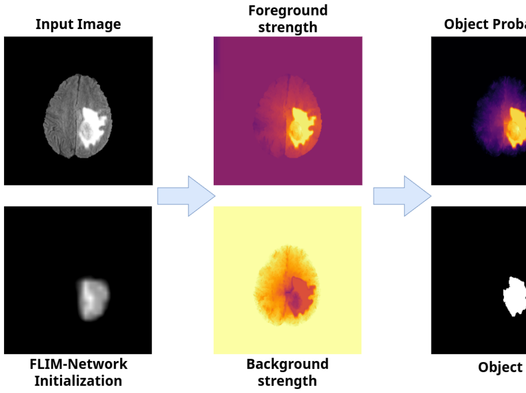


Fig. 3. The first column shows the input image and FLIM initialization to CA. The middle column shows the evolved tumor/foreground and background strength. The last column shows the tumor probability map and final saliency.

CA evolution. We also substituted $||\vec{I}(p)-\vec{I}(q)||_2$ by $\alpha||\vec{I}(p)-\vec{I}(q)||_2 + (1-\alpha)\Psi(\vec{J}(p),\vec{J}(q))$ in Equation 3, where Ψ can be the Euclidean or the cosine distance, to evaluate the use of FLIM features in our CA.

Algorithm 1 Cellular Automata initialized by FLIM

```
procedure CA($\mathbf{I}$, $\theta_l^0, l, L$)
    $t \leftarrow 0, dist \leftarrow +\infty$
    while $dist > 10^{-4}$ do
        $\theta_l^{t+1} \leftarrow \theta_l^t$                    ▷ Copy previous $\theta$
        for $\forall p \in D$ do
            $q_{max} \leftarrow \theta_l^t(p)$
            for $\forall q \in N(p)$ do                ▷ Moore
                $q_{aux} = g(p,q) \times \theta_l^t(q)$
                if $q_{aux} > q_{max}$ then
                    $\theta_l^{t+1}(p) \leftarrow q_{aux}$
                    $L(p) \leftarrow L(q)$
                    $q_{max} \leftarrow q_{aux}$
                end if
            end for
        end for
        $dist \leftarrow ||\theta_l^t - \theta_l^{t+1}||_2/|D_I|$
        $\theta_l^t \leftarrow \theta_l^{t+1}$                    ▷ Update step
        $t \leftarrow t+1$
    end while
    return $\theta_l^t$
end procedure
```

IV. Experimental Setup

We investigate our method on two medical datasets: (1) 2D axial slices of brain tumor from the BraTS 2021 dataset, three tumor slices (representing the median, 1^{st} and 3^{tr} quartis) for each data sample; (2) A private dataset of 2D parasite eggs, where each image may contain parasite eggs and impurities (Figure 4). BraTS 2021 has 3743 images, 1113 of which were separated for testing. The remaining 2630 were randomly divided into three split configurations, each with 5 training and 2625 validation images. Parasites have 1219 images, 366 of which comprised our test set, while 853 were randomly divided into three split configurations, each with five training and 848 validation images.

This decision simulates a scenario where abundant labeled data is not available. This is true for many practical problems, and our goal is to investigate and improve our method for problems where few or no labeled data is available. When no labeled data is available, a FLIM encoder could be designed by user drawings (weakly supervised).

We inserted foreground and background markers in discriminative regions of the five training images. To better control which image patches compose our encoder for tumors, we drew disks (defined by clicks). We design a CNN encoder with three convolutional layers (likewise a U-Net [6] encoder).

For tumors, we designed a FLIM encoder with kernels $3 \times 3 \times m$ in three layers with 16, 32, and 64 kernels. Each layer has marker-based normalization, convolution, ReLU, and max-pooling (3×3 and stride 2). We created a four-layer encoder for parasites with 27, 32, 32, and 16 kernels of size $3 \times 3 \times m$. We used the same operations on each layer, differing on max-pooling (3×3 and stride 1). We used dilation on convolutions with rates 3, 5, 7, and 7 to increase the receptive field.

FLIM-Encoder architectures were chosen based on experimental analysis. Exploring architectures is outside our scope, as our goal was to validate CA initialization through FLIM. The adaptive decoder was empirically configured with $A_1 = 0.1$ and $A_1 = 0.2$ for the area thresholds.

We initialized the background strengths θ_0^0 differently for each dataset. For BraTS, the brain mask is easily defined. We set $\theta_0^0 \leftarrow 1$ outside the brain and $\theta_0^0 \leftarrow 0$ inside the brain. For Parasites, we used θ_0^0 as the complement of the saliency map after a dilation of radius 10. θ_1^0 is the saliency map S (from a FLIM network or a DL-SOD) for both datasets. We also combined image properties and network features with multiple values of α and β, but for brevity, we report our best results only (with $\beta = 0.6$ and $\alpha = 0.7$). β improves cases where the tumor center is darker than the surrounding tumor (see Figures 2 and 5). Since parasites are darker than the background, we use $Y(p) < Y(q)$.

To compare our methods against DL-SOD methods, we integrated BasNET [3] and $\mathbf{U^2}$-Net [2] into our proposed pipeline. We used the pretrained models on the augmented DUTS-TR dataset (21106 images) and fine-tuned them using an empirically selected lower learning rate (1e-4).

We compared our results on three metrics: F-Score and μWF ($beta = 2$), and Dice. Those values range from 0% to 100%, and higher values mean a better saliency map.

V. Results & Discussions

Table I summarizes the results of CA initialization by a FLIM network for SOD. For BraTS [18], we seek the whole tumor detection, while for parasites, the parasite egg detection.

TABLE I
Results on Validation Sets

Tumors	F-Score	μWF	Dice
FLIM	51.1% ± 4.3%	41.7% ± 8.1%	47.8% ± 10.0%
FLIM_CA	**63.1% ± 0.7%**	**65.0% ± 0.6%**	**65.0% ± 0.4%**
Parasites	**F-Score**	**μWF**	**Dice**
FLIM	75.9% ± 1.5%	58.1% ± 1.4%	**63.5% ± 1.1%**
FLIM_CA	**81.0% ± 1.7%**	**58.6% ± 9.7%**	57.0% ± 8.5%

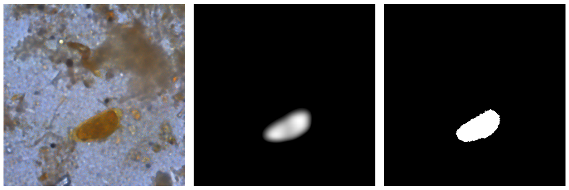
Initializing cell states with a FLIM network and further evolving them until convergence yields better saliency maps. We verify a substantial improvement for all metrics in the whole-tumor detection. Those improvements range from 12% to 23.3%. Likewise, it **stabilizes the models**, reducing the standard deviation across validation sets by 3.6%-9.6%. We verify that the ratio between precision and recall improved through an increase across all metrics and from the generated saliencies (see Figure 6b). Furthermore, our approach provides saliency distribution that better represents the ground truth.

TABLE II
BEST MODEL ON TEST SET

Tumors	F-Score	μWF	Dice
FLIM	56.2%	53.0%	61.9%
FLIM_CA	**63.3%**	**66.1%**	**65.9%**
Parasites	**F-Score**	**μWF**	**Dice**
FLIM	74.8%	64.8%	69.3%
FLIM_CA	**81.5%**	**72.4%**	**69.6%**

We also see a significant improvement in the F-Score for parasite egg saliency. However, the weighted F-Measure shows slight improvement, as the parasite eggs occupy a small portion of the image. We see a tangible deterioration of Dice for our saliency distribution. The source of this degeneration is two-fold: (1) contrary to tumors, eggs may appear anywhere in the image, so initializing the background seeds was a challenge (we use the complement of the FLIM-Network saliency map); (2) there are impurities similar to parasites identified as parasites, and our initialization generates false positives for impurities. We see instability across parasite validation sets, with higher standard deviations. The best model on validation improves all metrics on test for both datasets (Table II).

Figure 4 shows an example of our approach for parasite eggs. The improvement on edge regions is verified, where FLIM-Network provides lower initialization on high-frequency regions, and CA correctly fits the parasite region. Nevertheless, edge regions are noisy, also in tumor saliencies (Figure 6).



(a) Input Image (b) FLIM Saliency (c) CA Output

Fig. 4. Proposed method on Parasites.

We also investigated the evolution of the cell states using the FLIM-encoder features. For the sake of conciseness, we focused on tumors. Figure 5 shows one of the most substantial benefits of the FLIM Methodology. FLIM-designed CNN encoders show a delineation-to-detection tendency: shallow activations are suitable for saliency estimation, with sharp edges, though with some false positives; deeper activations, on the contrary, are suitable for object detection with fewer false positives and blurred edges.

Decoding the activations of a FLIM encoder results in a saliency suitable for initializing cell states. As edge regions are blurry, we initialize edge regions with lower strengths, which are improved during evolution. Initially, only image intensity was employed to evolve cell states. Here, we also used a feature vector for each pixel. We experiment with feature vectors representing activation channels of the first FLIM-encoder layer, as they show sharp edges.

Table III shows that evolution with features considerably degrades performance for all metrics. As the α parameter weights the contribution of image intensity and feature vectors, we verified that higher α yields better performances ($\alpha = 1$ results FLIM_CA). Given the initialization with blurrier edges, a guideline for future works is to start evolution using features from deeper layers and then move to the upper layer's features.

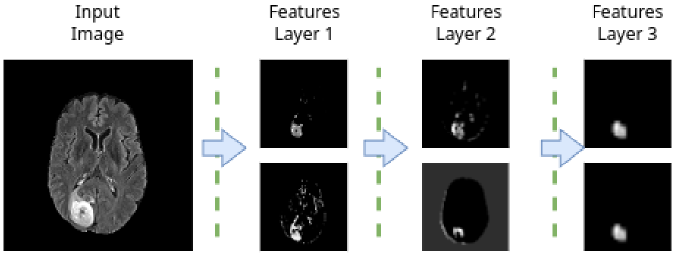


Fig. 5. Activations across the FLIM encoder

TABLE III
IMAGE'S INTENSITY AND FEATURES BASED CA (BRATS, $\alpha = 0.7$)

Validation	F-Score	μWF	Dice
FLIM_CA	**63.1% ± 0.7%**	**65.0% ± 0.6%**	**65.0% ± 0.4%**
FLIM_CAF*	56.9% ± 2.8%	57.1% ± 5.8%	58.7% ± 5.5%
FLIM_CAF†	29.5%±1.8%	38.9%±0.9%	52.3±0.5%
Test	**F-Score**	**μWF**	**Dice**
FLIM_CA	**63.3%**	**66.1%**	**65.9%**
FLIM_CAF*	60.2%	63.5%	64.5%
FLIM_CAF†	33.1%	40.5%	53.3%

* Euclidean Distance †Cosine Distance

To compute the similarity between the feature vectors, the results with Euclidean distance degrade less. Given a low-dimension feature vector (16 activation channels) and its non-sparsity, cosine similarity shows the worst results. Namely, the angles between close feature vectors are smaller, while the same does not hold in Euclidean space. Yet, deep-features CA with Euclidean distance improves FLIM saliencies.

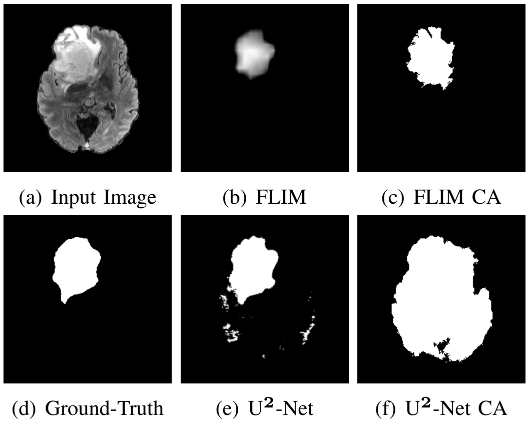
We investigated using saliency generated by deep-learning models (*i.e.,* BasNet and U^2-Net) to initialize the cell states for tumors. Table IV shows there is no improvement.

TABLE IV
CA INITIALIZED BY DEEP LEARNING MODELS (BRATS VALIDATION)

Basnet	F-Score	μWF	Dice
Basnet	**58.5% ± 7.3%**	**54.6% ± 4.4%**	**52.7% ± 7.3%**
Basnet_CA	10.3%±4.6%	22.5%±2.6%	34.5%±3.1%
U^2-Net	**F-Score**	**μWF**	**Dice**
U^2-Net	**64.8% ± 1.2%**	**62.6% ± 4.3%**	**64.8% ± 1.2%**
U^2-Net_CA	6.7%±2.8%	19.8%±4.1%	30.5%±5.8%

The observed results arise from two main problems: (1) Contrary to FLIM-Networks, models are too confident on edge regions (Figure 6d), even when wrong; (2) They also have many false positives with high confidence (*i.e.,* values close to 255). Hence, initializing the cell states under those conditions yields the worst saliency maps (Figure 6e).

It is valuable to point out that with a fraction of parameters, FLIM_CA shows comparable results with deep learning models under the restrictive scenario.



(a) Input Image (b) FLIM (c) FLIM CA

(d) Ground-Truth (e) U^2-Net (f) U^2-Net CA

Fig. 6. Comparison of CA initialized by FLIM againts U^2-Net.

VI. Conclusion & Future Work

We presented a solution for the initialization problem of CA that relies on the effectiveness of a FLIM network for SOD to set the CA's initial states. As an advantage, the method integrates expert knowledge during the initialization phase (the network design) and does not require user interaction in the validation and test phases. Our proposal reaches its goal of successfully improving FLIM-Network saliencies.

Our method was validated on public (BraTS) and private (Parasites) datasets. We show significant improvements, where CA improves FLIM saliency maps for both datasets. FLIM_CA shows comparative results to DL-SOD methods under labeled data constraints for the brain tumor dataset. Remarkably, such results were achieved with lightweight FLIM encoders and without backpropagation. Furthermore, we observed that DL-SOD methods under labeled data constraints are unsuitable for CA's initialization since false positives are only amplified, and models are too confident on edge regions. We also integrated the FLIM feature maps to guide the CA's evolution for evaluation. We noticed a drop in performance, which requires further investigation exploring features space.

For future work, evolving only one CA rather than two (background and foreground) may reduce computational time and simplify the design of evolution rules. It may allow more complex interactions with cells and facilitate the development of a 3D multi-label CA suitable for the tumor segmentation problem. Our implementation took advantage of CPU parallelization, and there is plenty of room for improvement through GPU parallelization. Moreover, we intend to study methods to smooth the generated saliency on the border's surface [13].

Acknowledgment

We thank the Eldorado Institute, CNPq (407242/2021-0, 306573/2022-9, 442950/2023-3, 304711/2023-3), FAPEMIG (APQ-01079-23, PCE-00417-24 and APQ-05058-23) and FAPESP (2023/14427-8 and 2013/07375-0), CAPES (STIC-AMSUD 88887.878869/2023-00).

References

[1] Y. Qin, M. Feng, H. Lu, and G. W. Cottrell, “Hierarchical cellular automata for visual saliency,” *International Journal of Computer Vision*, vol. 126, no. 7, pp. 751–770, Jul 2018.

[2] X. Qin, Z. Zhang, C. Huang, M. Dehghan, O. R. Zaiane, and M. Jagersand, “U2-net: Going deeper with nested u-structure for salient object detection,” *Pattern Recognition*, vol. 106, p. 107404, 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0031320320302077

[3] X. Qin, Z. Zhang, C. Huang, C. Gao, M. Dehghan, and M. Jagersand, “Basnet: Boundary-aware salient object detection,” in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[4] G. Zhu, L. Wang, and J. Tang, “Learning discriminative context for salient object detection,” *Engineering Applications of Artificial Intelligence*, vol. 131, p. 107820, 2024. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0952197623020043

[5] N. Tong, H. Lu, Y. Zhang, and X. Ruan, “Salient object detection via global and local cues,” *Pattern Recognition*, vol. 48, no. 10, pp. 3258–3267, 2015, discriminative Feature Learning from Big Data for Visual Recognition. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0031320314004932

[6] O. Ronneberger, P. Fischer, and T. Brox, “U-net: Convolutional networks for biomedical image segmentation,” in *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, N. Navab, J. Hornegger, W. M. Wells, and A. F. Frangi, Eds. Cham: Springer International Publishing, 2015, pp. 234–241.

[7] X. Zhou, Z. Li, and T. Tong, “Medical image segmentation and saliency detection through a novel color contextual extractor,” in *Artificial Neural Networks and Machine Learning – ICANN 2023*, L. Iliadis, A. Papaleonidas, P. Angelov, and C. Jayne, Eds. Cham: Springer Nature Switzerland, 2023, pp. 457–468.

[8] A.-I. Marinescu, Z. Balint, L.-S. Diosan, and A.-M. Andreica, “Unsupervised and fully autonomous 3d medical image segmentation based on grow cut,” in *2018 20th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing (SYNASC)*, 2018, pp. 401–408.

[9] J. von Neumann, “The general and logical theory of automata,” in *Cerebral mechanisms in behavior*, 1951, pp. 1–41.

[10] Y. Qin, H. Lu, Y. Xu, and H. Wang, “Saliency detection via cellular automata,” in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 110–119.

[11] V. Vezhnevets and V. Konushin, “”growcut” - interactive multi-label nd image segmentation by cellular automata,” *Graphicon*, vol. 1, 11 2004.

[12] A. Hamamci, N. Kucuk, K. Karaman, K. Engin, and G. Unal, “Tumor-cut: Segmentation of brain tumors on contrast enhanced mr images for radiosurgery applications,” *IEEE Transactions on Medical Imaging*, vol. 31, no. 3, pp. 790–804, 2012.

[13] C. Sompong and S. Wongthanavasu, “An efficient brain tumor segmentation based on cellular automata and improved tumor-cut algorithm,” *Expert Systems with Applications*, vol. 72, pp. 231–244, 2017. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0957417416305772

[14] I. E. de Souza and A. X. Falcão, “Learning cnn filters from user-drawn image markers for coconut-tree image classification,” *IEEE Geoscience and Remote Sensing Letters*, vol. 19, pp. 1–5, 2022.

[15] I. E. de Souza, B. C. Benato, and A. X. Falcão, “Feature learning from image markers for object delineation,” in *2020 33rd SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI)*, 2020, pp. 116–123.

[16] L. d. M. Joao, B. M. d. Santos, S. J. F. Guimaraes, J. F. Gomes, E. Kijak, A. X. Falcao *et al.*, “A flyweight cnn with adaptive decoder for schistosoma mansoni egg detection,” *arXiv preprint arXiv:2306.14840*, 2023.

[17] L. Joao., M. Cerqueira., B. Benato., and A. Falcao., “Understanding marker-based normalization for flim networks,” in *Proceedings of the 19th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications - Volume 2: VISAPP*, INSTICC. SciTePress, 2024, pp. 612–623.

[18] U. Baid *et al.*, “The RSNA-ASNR-MICCAI BraTS 2021 Benchmark on Brain Tumor Segmentation and Radiogenomic Classification,” *arXiv e-prints*, p. arXiv:2107.02314, Jul. 2021.