AutoFLIM: Joint Optimization of Architecture, Markers, and Hyperparameters for Efficient Medical Image Segmentation

Gilson J. Soares[1(✉)][0000−0001−8805−5644], Felipe C. R. Salvagnini[1][0000−0002−7896−0058], Jeova F. S. R. Neto[2][0000−0001−8103−7937], and Alexandre X. Falcão[1][0000−0002−2914−5380]

[1] Institute of Computing, University of Campinas, Campinas, SP, Brazil
{g272446,f202285}@students.ic.unicamp.br, afalcao@unicamp.br
[2] Bowdoin College, Department of Computer Science, Brunswick, ME, USA
j.farias@bowdoin.edu

Abstract. Modern deep learning methods for medical image segmentation require large training datasets and substantial computational power for both training and deployment. This could prevent the use of these methods in domains such as parasitology, where only a few annotated datasets are available, and in regions with limited access to computational resources. Previous work introduces Feature Learning from Image Markers (FLIM) as a methodology for training CNN encoders using only patches extracted from marker annotations of a few representative images. An advantage of this method is that it requires less data and can achieve close to state-of-the-art results with small networks. Despite these advances, prior approaches still rely on users for image selection, marker annotation, and architecture definition, constraining scalability. In this work, we overcome these limitations with a methodology for jointly optimizing the architecture, markers, and hyperparameters, which we call AutoFLIM. The method uses a Genetic Algorithm with a multi-objective formulation that balances segmentation (Dice coefficient) and computational efficiency (FLOPs). Because FLIM training requires no backpropagation, the search explores a far larger space than a grid search or manual design. We evaluate our approach on two parasitology microscopy datasets, demonstrating that competitive segmentation can be achieved with only a few training images, approaching the performance of more computationally intensive approaches in the low-data regime, while using $322 - 558\times$ fewer parameters and $14 - 42\times$ fewer FLOPs. The final encoder is built from only 4–5 marked images, showing that few images suffice to train a competitive model; however, the greedy selection step still scores candidates against the annotated pool, so the current annotation budget remains that of the full labeled set. Removing this dependence is the natural next step. The code is available at https://github.com/LIDS-UNICAMP/AutoFLIM.

Keywords: Architecture search · Hyperparameter optimization · Image selection · Medical image segmentation

1 Introduction

Despite impressive progress in medical image segmentation, applying state-of-the-art methods in constrained environments remains challenging. The need for large volumes of data, annotation, and computational resources poses a challenge for specific domains, such as parasitology, which is especially important in developing countries. These requirements manifest across the main families of segmentation methods, each of which depends on dense annotation to some degree. Supervised methods, such as the classical U-Net [15], can be applied directly when annotated data is available. Modern methods such as nnU-Net [9] serve as a self-configuring framework that automatically adapts its hyperparameters and network architecture to the dataset fingerprint and hardware capabilities. More recent architectures such as CMUNeXt [21] target efficiency directly, achieving competitive accuracy with far fewer parameters; however, like other supervised approaches, they still depend on dense pixel-level annotation. Self-supervised methods such as LeJEPA [1] and DINO [3] train a backbone on unlabeled data but still require annotations for the downstream task, and foundation models such as SAM [11] do not transfer to specialized domains without annotated fine-tuning, as in MedSAM [13].

An alternative direction avoids auxiliary datasets entirely. The FLIM (Feature Learning from Image Markers) framework [6] trains encoders from weakly annotated images using image markers, combined with a heuristic Adaptive Decoder [10,17,18,19], achieving competitive segmentation results with minimal supervision [17,18,19]. Object delineation is further improved [17,18] by graph-based methods such as Dynamic Trees (DT) [2]. However, most of these works use fixed architectures, manually setting the number of layers and other network hyperparameters. The work [18] optimizes only the number of layers, and in [14] the authors perform a grid search over FLIM parameters; in both, the annotation step still relies on human selection. Neither leverages the rich body of neural architecture search (NAS) [16], which is designed for gradient-trained networks and therefore not directly applicable to FLIM's backprop-free encoders, motivating a search strategy, such as ours, compatible with this setting. This human-in-the-loop design [4,5] introduces variance from differing user judgment, which prior work mitigates only by averaging over multiple annotators [18], time-consuming and prohibitive for rapid experimentation.

To address these limitations, we propose AutoFLIM, a supervised method that jointly and automatically searches over both the encoder architecture and the subset of training images within the FLIM framework. Taking advantage of FLIM's fast training and inference, we formulate a multi-objective optimization problem using a genetic algorithm with Dice and FLOPs as objectives. Keeping image selection separate from the network chromosome preserves annotation efficiency and framing, and avoids inflating the search space. Because training requires no gradient descent, the resulting encoders are extremely compact and fast at inference, making the method well-suited to low-resource clinical settings. This work shows that the final encoder can be trained with 4–5 marked images; however, the greedy selection procedure scores each candidate against its ground

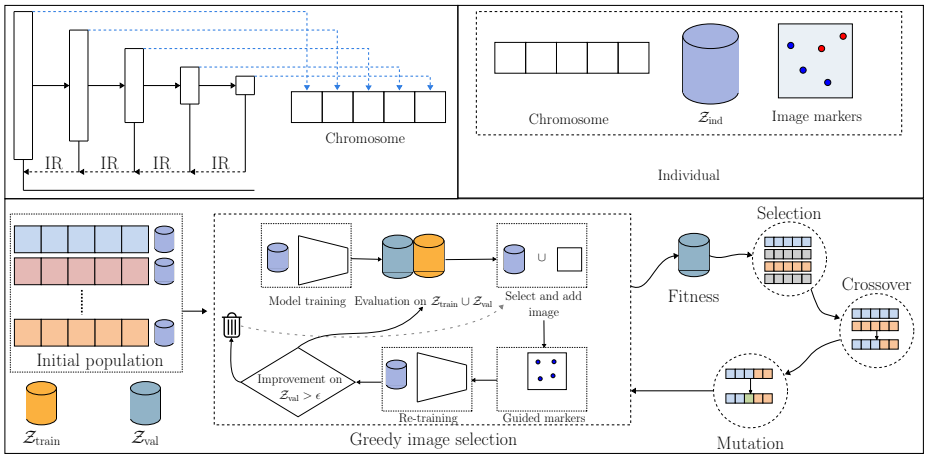

Fig. 1. Method overview: Each FLIM architecture is encoded as a chromosome, with IR showing the inferior reconstruction used as the model's final output; an individual combines the chromosome, its training images set $\mathcal{Z}_{\text{ind}}$, and the image markers. Each undergoes greedy image selection, adding images from $\mathcal{Z}_{\text{train}}$ while fitness on $\mathcal{Z}_{\text{val}}$ improves and drives the GA.

truth, which causes the selection to consume the annotation budget of the labeled pool during search. We demonstrate that a few images are sufficient to train a competitive model.

Our major contributions to this work can be summarized as: (I) a genetic-algorithm-based method that jointly searches encoder architecture and training image subset with the FLIM framework, removing the need for expert judgment in image selection, marker placement, and architecture design; (II) within the genetic algorithm optimization, we develop a method for automatically marking selected images from the model's error regions, while the choice of which images to annotate is still guided by the labeled pool; (III) evaluation across prototypical datasets, showing that the method is competitive under scarcity of both annotated and computational resources.

2 Methodology

The proposed method, illustrated in Figure 1, consists of optimizing both the architecture and the images and their markers. The FLIM architecture is encoded as a chromosome, while the individual is represented by the set comprising the chromosome, the training image set, and the image markers. The AutoFLIM method is shown in the second part of the figure: it starts with an initial population of individuals, each one with a random start image sampled from $\mathcal{Z}_{\text{train}}$. The process consists of training a FLIM encoder with it, evaluating on $\mathcal{Z}_{\text{train}}$ and $\mathcal{Z}_{\text{val}}$, selecting a new image from $\mathcal{Z}_{\text{train}}$, and adding it to the individual training

Algorithm 1: Multi-objective memetic search for FLIM encoders

Input : search space Ω; train pool $\mathcal{Z}_{\text{train}}$, validation set $\mathcal{Z}_{\text{val}}$; population μ, generations G, mutation rate p_m, stagnation patience P
Output: Pareto front $\mathcal{F}$ over (Dice↑, FLOPs↓); best encoder $\gamma^\star$

$\mathcal{P} \leftarrow \mu$ random genomes, each seeded with one random image from $\mathcal{Z}_{\text{train}}$
score every $\gamma \in \mathcal{P}$ with `GreedyFit` → (Dice, FLOPs)
$\mathcal{F} \leftarrow$ non-dominated front of $\mathcal{P}$; $\gamma^\star \leftarrow \arg\max_{\gamma \in \mathcal{F}}$ Dice; $c_{\text{stag}} \leftarrow 0$
for $g \leftarrow 1$ **to** G **do**
  form μ unique, valid offspring: crowded binary tournament → depth-aware uniform crossover → per-gene mutation (p_m), each re-seeded with a fresh random image
  score every offspring with `GreedyFit`
  $\mathcal{P} \leftarrow$ `NSGA2(`$\mathcal{P} \cup$ *offspring*, μ`)` `// elitist: best` μ `of` 2μ `by non-dom. sort + crowding`
  $\mathcal{F} \leftarrow$ non-dominated front of $\mathcal{P}$
  if $\max_{\gamma \in \mathcal{F}}$ *Dice* > *Dice*$(\gamma^\star)$ **then**
    $\gamma^\star \leftarrow \arg\max_{\gamma \in \mathcal{F}}$ Dice; $c_{\text{stag}} \leftarrow 0$
  else
    c_{stag} += 1; **if** $c_{stag} \geq P$ **then break**
  end
end
return $\mathcal{F}$, *and* $\gamma^\star$ *retrained on its selected images with the refined markers*

set $\mathcal{Z}_{\text{ind}}$. The model's error regions guide image annotation: if the improvement exceeds the threshold ϵ, the image is kept; otherwise, it is discarded, and a new image is selected. Outside the greedy image selection, the steps of the genetic algorithm- selection, crossover, and mutation are realized using the fitness computed on the validation sets. The full procedure is summarized in Algorithm 1. We use a population of $\mu = 30$ individuals evolved over $G = 40$ generations, with binary tournament selection (size 2) and a stagnation patience of $P = 20$ generations.

2.1 FLIM

Images and markers FLIM estimates convolutional kernels from the $k \times k$ patches centered on marker pixels [6]; both the marker radius ρ and the kernel size k are searched rather than fixed (Table 1).

FLIM encoder The encoder has B convolutional blocks whose kernels are estimated directly from the marker patches. We adopt the Bag of Feature Points variant [14]: for each marker s the patches $\mathcal{P}_s$ are clustered independently and the cluster centers define that block's kernels.

FLIM decoder We use the fixed mean-based adaptive decoder of [18]. The per-layer saliency maps produced by the decoder are then merged by *inferior*

Table 1. Parameters used for the genes of the chromosomes in the search space.

Gene	Values
Number of layers (L)	$\{2, 3, 4, 5, 6\}$
Kernel size (k_{L_i})	$\{3, 5, 7, 9\}$
Pooling type (π_{L_i})	{stride-1 conv, stride-2 conv, avg-pool, max-pool}
Marker radius (ρ)	$\{1.5, 3.0, 6.0\}$

reconstruction (reconstruction by erosion) [8] to generate the final segmentation. The decoder and this combination are held fixed across all experiments. Different from previous works [18], we do not use a graph-based algorithm to further refine the output, since we aim to keep all the processing in GPU.

2.2 Genetic Algorithm

Chromosome and search space The chromosome that represents each FLIM architecture is composed of the following genes: number of layers (L), kernel size (k_{L_i}), pooling type (π_{L_i}), marker size (circular radius ρ). Some fixed hyper-parameters are also used like maximum number of channels (500), number of kernels per marker (1), activation function (ReLU), type of adaptive decoder (mean-based) and combination of layers (inferior reconstruction). Table 1 shows the values used on the search space.

Genetic operators *Selection*: The selection is done by first using the binary tournament of NSGA-II [7], called the crowded-comparison operator: the individual with lower Pareto rank wins, with ties being broken by the larger crowding distance. Survivors are then selected for the next generation using an elitist plus-selection scheme: the μ parents and their μ offspring are merged into a pool of 2μ individuals, sorted into non-dominated fronts, and the next population keeps the best μ front by front, truncating the last admitted front by descending crowding distance. Elitism guarantees that the best encoder never regresses across generations.

Crossover: Chromosomes of different individuals could have different depths, which forces our algorithm to use a depth-aware uniform crossover. The scalar genes, such as the number of layers, are inherited independently from either parent with probability of 0.5; then, for each active layer $i \leq L$, the pair (k_{L_i}, π_{L_i}) is copied as a unit from a parent that has a real gene at that depth. This always yields a valid child, whereas plain per-gene produces invalid layers when the parents differ in depth.

Mutation: Each gene of the individual chromosome is independently resampled uniformly from the allowed values with probability p_m=0.3.

Greedy image selection (GreedyFit) Each chromosome γ is scored by a greedy procedure that jointly grows the encoder's training image set and its

markers. Starting from a single random image marked with random ground-truth foreground/background points ($n_{\text{fg}} = n_{\text{bg}} = 5$) we train a FLIM encoder and then, for up to T=5 steps, add the train-pool image whose current Dice lies in the 25–50th percentile (harder than the median, yet not a lowest-Dice outlier). The candidate is marked at the model's error regions (foreground markers on false negatives, background on false positives), the encoder is retrained, and the image is *accepted if the mean validation Dice improves by at least* ε=0.025; the procedure stops after r=3 consecutive rejections. The individual's fitness is the pair (Dice on $\mathcal{Z}_{\text{val}}$, FLOPs) of the resulting model, and its selected images and refined markers are retained for final training. Greedy selection evaluates each candidate on the annotated pool, so the annotation cost is presently that of the full labeled set, even though the final model trains on only a few images. Reducing this cost is a target for future work.

3 Experiments and Results

Datasets The method was evaluated on two different parasitology datasets: S. mansoni [12], which contains 1219 images of eggs of *Schistosoma mansoni* with size 400 × 400; Larvae [20], which contains 494 images of larvae of the species *Strongyloides stercoralis* of size 256. Originally, the larvae dataset had a major imbalance, with images containing only the background; for this study, we created a balanced dataset with the same number of images with and without larvae. Images needed to be padded and resized to have a size of 256. For all datasets, we use the same split method: 50% of the data is held out for testing, and the remaining 50% form the training pool, from which 10% is reserved for validation (resulting in 549 training and 60 validation images for S. mansoni, and 222 and 24 for Larvae) . This process is repeated three times to reduce the possibility of selecting a good set by chance. All our results present the mean and standard deviation across these three splits.

Comparative methods We compare our method with two models: nnU-Net and CMUNeXt, two well-known works widely used in medical imaging. Both were trained with default configurations and hyperparameters, with only the epochs for nnU-Net being adjusted to 100 and trained in just one fold. For fairness of results, all post-processings applied to FLIM results, such as area filters, were also applied to the comparative methods with same area sizes computed from the training images. A single AutoFLIM search takes approximately 138 minutes on S. mansoni and 24 minutes on Larvae on one RTX 5080. For reference, training nnU-Net on the same data takes 50 min and 13 min, and CMUNeXt takes 19 min and 8 min. Note that our method optimizes both the architecture and markers and searches for images, while the baselines train a single configuration without selecting images. The time reported here is a one-time offline cost, distinct from the inference cost in GFLOPs reported in Table 2. Replacing greedy selection with a label-free alternative would remove much of this cost, since selection currently accounts for approximately half of the runtime.

Table 2. Model size, cost, and marker-image budget across datasets.

Method	S.mansoni			Larvae		
	# Img.	# Params (M)	GFLOPs	# Img.	# Params (M)	GFLOPs
FLIM-based	4 ± 1	0.104 ± 0.044	2.18 ± 0.72	5 ± 1	0.060 ± 0.014	2.14 ± 2.50
nnU-Net	549 ± 0	33.472 ± 0.000	90.65 ± 0.00	222 ± 0	33.472 ± 0.000	29.60 ± 0.00
CMUNeXt	549 ± 0	3.149 ± 0.000	14.46 ± 0.00	222 ± 0	3.149 ± 0.000	14.46 ± 0.00

Table 3. Performance comparison across datasets using Dice, IoU, and MAE.

Method	S.mansoni			Larvae		
	Dice ↑	IoU ↑	MAE ↓	Dice ↑	IoU ↑	MAE ↓
FLIM-based	0.794 ± 0.005	0.754 ± 0.004	0.007 ± 0.000	0.623 ± 0.038	0.587 ± 0.039	0.022 ± 0.004
nnU-Net	0.946 ± 0.007	0.924 ± 0.007	0.002 ± 0.000	0.921 ± 0.002	0.902 ± 0.002	0.004 ± 0.000
CMUNeXt	0.942 ± 0.001	0.917 ± 0.001	0.002 ± 0.000	0.922 ± 0.010	0.902 ± 0.009	0.005 ± 0.000

3.1 Results

Table 2 highlights the core advantage of FLIM: it uses 322× and 558× fewer parameters than nnU-Net on S. mansoni and Larvae, respectively (30× to 52× fewer than CMUNeXt), as well as 42× and 14× fewer GFLOPs on the same hardware, and it constructs the network weights from only a few images per dataset rather than the full training set. This shows that our method consistently finds a compact core set that achieves competitive accuracy at a fraction of the computational budget. Crucially, the size of this accuracy cost depends on the annotation regime: as Figure 2 shows, FLIM stays close to the supervised baselines in the low-data regime, but the gap widens substantially when the baselines have the full training set.

The per-dataset scores in Table 3 report this upper bound: with the full training set, nnU-Net reaches 0.946 Dice on S. mansoni and 0.921 on Larvae, against 0.794 and 0.623 for FLIM trained on 4 ± 1 and 5 ± 1 images. In exchange, the searched encoders use 322× and 558× fewer parameters than nnU-Net. Whether this trade is acceptable depends on the task, as it favors settings where computa-

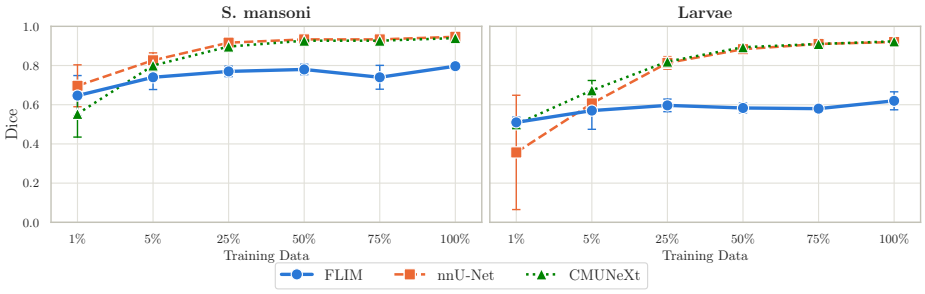

Fig. 2. Results on different regimes of training data for all datasets. The mean Dice results are shown, with error bars displaying the standard deviation across splits.

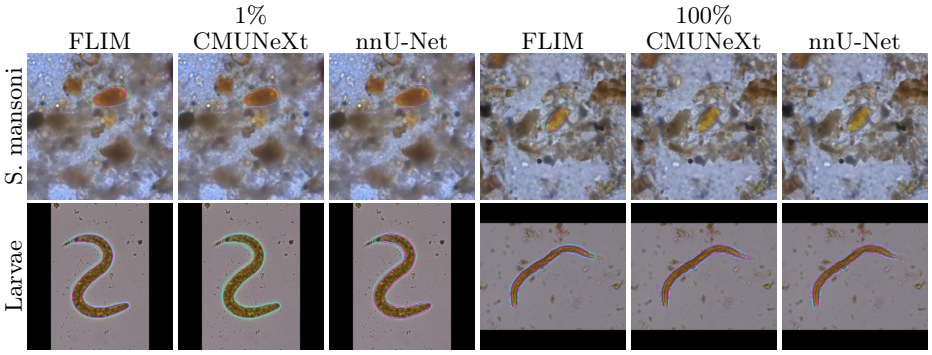

Fig. 3. Qualitative examples at 1% and 100% of the training data, with ground-truth contours overlaid in cyan and predictions in magenta. At 1%, both FLIM and the baselines produce similar results, with CMUNeXt completely failing on the Larvae segmentation. At 100%, the baselines correctly segment both datasets, while FLIM still segments but loses on border segmentation.

tional resources at deployment are the binding constraints and could be a strong alternative in pipelines with further segmentation refinements. The capacity of our method to find compact core sets of training images and architectures across regimes is highlighted in Figure 2. FLIM remains stable across all percentages, whereas the comparative methods degrade sharply as training data is reduced. As expected, the baselines improve with more data, since backpropagation requires larger training sets to perform well; however, this accuracy comes at the cost of far heavier models. Figure 3 shows qualitative examples at both extremes of the annotation range. At 1%, the baselines and FLIM achieve similar segmentation results, though some baselines fail to learn with this small amount of data (CMUNeXt on Larvae). At 100% of the training data, the baselines produce accurate segmentations, with FLIM still correctly finding the objects but with less well-defined borders.

Because the search jointly optimizes Dice and FLOPs, it returns a non-dominated set of encoders instead of a single model. The Figure 4 shows the fronts obtained on each split. On S. mansoni, the fronts span 0.04 to 2.90 GFLOPs per image and follow a consistent shape across all splits, differing by a median of 0.08 at matched compute, showing that the search is stable across the different data partitions. On Larvae the front spans 0.02 to 5.67 GFLOPS per image. In this work, we report the maximum-Dice member, but cheaper encoders are available without rerunning the search, which is useful when deployment compute is a constraint.

3.2 Ablation

To validate the mechanisms of our method, a set of ablation studies was conducted, isolating each part of the method and evaluating its impacts on the

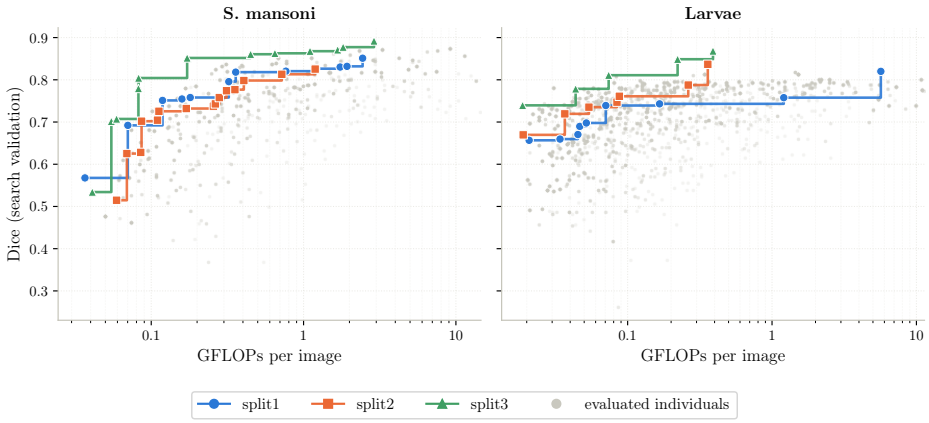

Fig. 4. Dice/FLOP fronts from the search, one front per split. Grey points are all evaluated individuals. The coloured markers trace each non-dominated front of each split. We compute Dice on the validation set during the search, so these values are not comparable to Table 3.

results: (a) Image selection, using random and greedy selection; (b) Image marking, using random and inference-guided markers; and (c) Architecture search via GA algorithm, using a fixed architecture in the place of architecture search. The fixed architecture uses 2 layers, with a kernel size of 3 (as used in [18]). The results are shown in Table 4. For S. mansoni, the single use of greedy selection does not improve Dice (0.754 to 0.746), as the placement of random markers can introduce noise during image search. This is evident in the standard deviation of the first two ablations. When inference-guided marking was introduced, it improved the results to 0.781, with a small standard deviation, showing an improvement even with a random selection of training images. The combination of greedy selection and inference-guided marking improves the results even further (0.794). Using a fixed architecture while keeping the greedy selection and guided markers loses 3.9 Dice points compared with the GA, demonstrating the impact of our method. The same ordering holds on Larvae, where the fixed architecture again lowers the result (0.623 to 0.572), confirming that the architecture search contributes across both datasets.

Table 4. Ablation study: contribution of smart selection and inference-guided markers.

Method	S.mansoni			Larvae		
	Dice ↑	IoU ↑	MAE ↓	Dice ↑	IoU ↑	MAE ↓
Rand. sel. + Rand. markers	0.754 ± 0.026	0.715 ± 0.025	0.008 ± 0.001	0.566 ± 0.010	0.524 ± 0.008	0.026 ± 0.000
Greedy sel. + Rand. markers	0.746 ± 0.056	0.707 ± 0.056	0.009 ± 0.003	0.592 ± 0.011	0.551 ± 0.012	0.025 ± 0.002
Rand. sel. + Inf.-guided markers	0.781 ± 0.004	0.739 ± 0.005	0.008 ± 0.000	0.596 ± 0.033	0.554 ± 0.032	0.023 ± 0.002
Greedy sel. + Inf.-guided markers	0.794 ± 0.005	0.754 ± 0.004	0.007 ± 0.000	0.623 ± 0.038	0.587 ± 0.039	0.022 ± 0.004
Fixed arch + Greedy sel. + Inf.-guided markers	0.755 ± 0.032	0.710 ± 0.027	0.009 ± 0.001	0.572 ± 0.013	0.523 ± 0.013	0.026 ± 0.002

4 Conclusion

In this work, we propose AutoFLIM, a method that automates the FLIM pipeline by jointly searching for the network architecture via a genetic algorithm and selecting an image set greedily, with markers automatically generated from the model's error regions. We demonstrated that, in low-data regimes, our method achieves competitive results relative to the backpropagation-based baselines while using significantly fewer parameters. Our ablation results show that the components interact and contribute to the result, with inference-guided marking making greedy selection pay off, while replacing the searched architecture with a fixed one costs 3.9 to 5.1 Dice points. We have shown that a competitive segmentation model can be constructed from only a few training images; however, the current selection step still requires the fully annotated pool. The clear path forward is annotation-free selection: an unsupervised or self-supervised replacement for greedy selection would turn the few-image capability shown here into few-annotation practice. Beyond this, future work could explore other search strategies and improved network building blocks.

Acknowledgments. The authors thank the São Paulo Research Foundation (FAPESP, #2025/19856-0, #2023/14427-8, and #2025/23020-4) and the Brazilian National Council for Scientific and Technological Development (CNPq, #304711/2023-3 and #408003/2025-1) for the financial support.

Disclosure of Interests. The authors have no competing interests to declare that are relevant to the content of this article.

References

1. Balestriero, R., LeCun, Y.: Lejepa: Provable and scalable self-supervised learning without the heuristics. arXiv preprint arXiv:2511.08544 (2025)
2. Bragantini, J., Martins, S.B., Castelo-Fernandez, C., Falcão, A.X.: Graph-based image segmentation using dynamic trees. In: Iberoamerican Congress on Pattern Recognition. pp. 470–478. Springer (2018). https://doi.org/10.1007/978-3-030-13469-3_55
3. Caron, M., Touvron, H., Misra, I., Jégou, H., Mairal, J., Bojanowski, P., Joulin, A.: Emerging properties in self-supervised vision transformers. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 9650–9660 (2021)
4. Cerqueira, M.A., Sprenger, F., Teixeira, B.C., Falcão, A.X.: Interactive image selection and training for brain tumor segmentation network. In: 2024 46th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC). pp. 1–4. IEEE (2024). https://doi.org/10.1109/EMBC53108.2024.10781962
5. Cerqueira, M.A., Sprenger, F., Teixeira, B.C., Guimarães, S.J.F., Falcão, A.X.: Interactive ground-truth-free image selection for flim segmentation encoders. In: 2024 37th SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI). pp. 1–6. IEEE (2024). https://doi.org/10.1109/SIBGRAPI62404.2024.10716300

6. De Souza, I.E., Falcão, A.X.: Learning cnn filters from user-drawn image markers for coconut-tree image classification. IEEE Geoscience and Remote Sensing Letters **19**, 1–5 (2020). https://doi.org/10.1109/LGRS.2020.3020098
7. Deb, K., Pratap, A., Agarwal, S., Meyarivan, T.: A fast and elitist multiobjective genetic algorithm: Nsga-ii. IEEE transactions on evolutionary computation **6**(2), 182–197 (2002). https://doi.org/10.1109/4235.996017
8. Falcao, A.X., Stolfi, J., de Alencar Lotufo, R.: The image foresting transform: Theory, algorithms, and applications. IEEE transactions on pattern analysis and machine intelligence **26**(1), 19–29 (2004). https://doi.org/10.1109/TPAMI.2004.1261076
9. Isensee, F., Jaeger, P.F., Kohl, S.A., Petersen, J., Maier-Hein, K.H.: nnu-net: a self-configuring method for deep learning-based biomedical image segmentation. Nature methods **18**(2), 203–211 (2021). https://doi.org/10.1038/s41592-020-01008-z
10. Joao., L., Cerqueira., M., Benato., B., Falcão., A.: 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. pp. 612–623. INSTICC, SciTePress (2024). https://doi.org/10.5220/0012385900003660
11. Kirillov, A., Mintun, E., Ravi, N., Mao, H., Rolland, C., Gustafson, L., Xiao, T., Whitehead, S., Berg, A.C., Lo, W.Y., et al.: Segment anything. In: 2023 IEEE/CVF international conference on computer vision (ICCV). pp. 3992–4003. IEEE (2023). https://doi.org/10.1109/ICCV51070.2023.00371
12. Laboratory of Image Data Science (LIDS), University of Campinas: Dataset of fecal microscopy images containing *schistosoma mansoni* eggs. https://github.com/LIDS-Datasets/schistossoma-eggs (2024), accessed: 2026-08-24
13. Ma, J., He, Y., Li, F., Han, L., You, C., Wang, B.: Segment anything in medical images. Nature communications **15**(1), 654 (2024). https://doi.org/10.1038/s41467-024-44824-z
14. Martinelli, J.D., Rodrigues Filho, M.L., Salvagnini, F.C.d.R., Soares, G.J., Santos, J.A.d., Falcão, A.X.: Flim networks with bag of feature points. In: Iberoamerican Congress on Pattern Recognition. pp. 268–281. Springer (2025). https://doi.org/10.1007/978-3-032-23176-5_19
15. Ronneberger, O., Fischer, P., Brox, T.: U-net: Convolutional networks for biomedical image segmentation. In: Medical image computing and computer-assisted intervention–MICCAI 2015: 18th international conference, Munich, Germany, October 5-9, 2015, proceedings, part III 18. pp. 234–241. Springer (2015). https://doi.org/10.1007/978-3-319-24574-4_28
16. Salmani Pour Avval, S., Eskue, N.D., Groves, R.M., Yaghoubi, V.: Systematic review on neural architecture search. Artificial Intelligence Review **58**(3), 73 (2025). https://doi.org/10.1007/s10462-024-11058-w
17. Salvagnini, F.C.R., Gomes, J.F., Santos, C.A., Guimarães, S.J.F., Falcão, A.X.: Improving flim-based salient object detection networks with cellular automata. In: 2024 37th SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI). pp. 1–6. IEEE (2024). https://doi.org/10.1109/SIBGRAPI62404.2024.10716266
18. Soares, G.J., Cerqueira, M.A., Gomes, J.F., Najman, L., Guimarães, S.J.F., Falcão, A.X.: Flim-based salient object detection networks with adaptive decoders. Journal of the Brazilian Computer Society **32**(1), 750–765 (Apr 2026). https://doi.org/10.5753/jbcs.2026.5893
19. Soares, G.J., Cerqueira, M.A., Guimaraes, S.J.F., Gomes, J.F., Falcão, A.X.: Adaptive decoders for flim-based salient object detection networks. In: 2024 37th SIB-

GRAPI Conference on Graphics, Patterns and Images (SIBGRAPI). pp. 1–6. IEEE (2024). https://doi.org/10.1109/SIBGRAPI62404.2024.10716333
20. Suzuki, C.T., Gomes, J.F., Falcao, A.X., Shimizu, S.H., Papa, J.P.: Automated diagnosis of human intestinal parasites using optical microscopy images. In: 2013 IEEE 10th international symposium on biomedical imaging. pp. 460–463. IEEE (2013). https://doi.org/10.1109/ISBI.2013.6556511
21. Tang, F., Ding, J., Quan, Q., Wang, L., Ning, C., Zhou, S.K.: Cmunext: An efficient medical image segmentation network based on large kernel and skip fusion. In: 2024 IEEE International Symposium on Biomedical Imaging (ISBI). pp. 1–5. IEEE (2024). https://doi.org/10.1109/ISBI56570.2024.10635609