Actin-Villin Fiber Segmentation: Use ML and Statistical Tests For Missing Fiber Models ? A Case Study from Cryo-EM tomography

Sotirios N. Raptis

Abstract—**This paper addresses the longstanding challenge of distinguishing relevant regions from background in images without prior knowledge, leveraging class statistical separability alongside three machine learning methods. The approach is demonstrated on a Cryo-EM molecular tomography scan of actin–villin fibers using three classification techniques: Logistic Regression for binary outcomes, K-Means for basic clustering, and hierarchical clustering for flexible merging. The methods are evaluated for consistency across varying classification parameters and validated using statistical tests. The analyzed image exhibits moderate texture with elongated segments that intermix but can be further partitioned into multiple clusters. Segmentation results are internally validated by varying execution parameters, demonstrating the robustness of the approach. These methods highlight applications in unsupervised segmentation and can be extended to segment videos or image sequences with minimal input.**

Index Terms—**Background, foreground, Kolmogorov Smirnov test, Kullback Leibler divergence, ROC analysis, Hierarchical Clustering, K-Means**

I. INTRODUCTION

One can divide almost any image into interesting and non-interesting regions, roughly termed foreground or background, which may contain objects spanning multiple image classes. This paper addresses the foreground / background separation problem as a dual-class task and proposes that the lack of prior knowledge can be mitigated by using statistical criteria to naturally separate image blocks.

Defining foreground and background largely involves reducing the ambiguity of certain pixels or blocks, a topic widely discussed in the literature. This ambiguity arises from background (background) and foreground (foreground) overlap, as these major regions are often ill-defined. The present work relies on the self-consistency of dual-class segmentation results, checking whether the two basic classes produce consistent pixel maps of foreground (FG) pixels. A first step toward assessing self-consistency is using different cluster counts and varying block sizes. Seed pixels can also be used as a priori patterns of foreground or background and then applied in segmentation based on similarity to these patterns. This approach, known as pixel annotation, does not introduce a significant Ground Truth (GT) bias due to the limited number of GT pixels used—no more than 10 in this work. Producing statistically derived FG/BG maps for comparison can lead to reliable segmentation results.

The core idea is to use statistical hypothesis testing to compare both the maps produced and neighboring classified blocks after applying a major segmentation algorithm that does not rely heavily on prior annotation. The KL and KS tests are examples of such methods, testing the similarity of neighboring blocks. Standard statistical models, such as the normal distribution ($N(0,1)$), are used to approximate similarity statistics. This allows computation of $p-value$ or the significance of block similarity (FG or BG), an assumption widely accepted in statistical learning.

No theoretical separation exists between foreground and background; this distinction is instead addressed through self-consistency tests and multiclass merging. The foreground and background pixels constitute broad pixel classes, with distinctions depending on the application [Dong2018SingleLabelMI(2018)]. Their success relies on how these image parts are repeated when parameters capturing either foreground or background change [DebaniPrasadMishra(2024)]. Consequently, foreground and background can span more specific content-related classes, forming a multiclass problem. The present work processes Cryo-EM 2D projection images, as in Fig. 2a (denoted as *'AV'*) from [HamptonCM(2008)], [av(2008)].

In its simplest form, foreground / background separation depends on salient geometrical features, such as well-defined objects with clear edges. Vague or missing edges often prevent exact foreground and background definitions.

Both foreground and background can consist of complex components, as discussed in [Phalempin(2021)]. Often, the background is merely noise, which can be removed through noise modeling, as in Fig. 2a. Rich backgrounds with high pixel depth (many gray levels per pixel) can produce complex background difficult to model. High pixel depth can degrade edges (many small pixel differences reduce the sharpness of major edges) and create high detail, producing new edges that reveal hidden structures. Regression (averaging) is used to smooth the background. Linear regression models predict pixel values via linear extrapolation from neighboring pixels, producing a smoother image and flattening fine signal variations. Robust edge detection and removal of low-intensity edges (mainly noise) further smooth the background. A rich background produces many low-intensity edges after edge detection. Edge parameters, such as pixel differences and

Dr. Raptis is an Alumnus with the Department of Informatics, National Technical University of Athens(NTUA), Athens, Greece (e-mail: sotiris.raptis.n@gmail.com

Manuscript received 3, 2026; revised XX/XX, 2026.

averaging spans, can be adjusted to remove scattered small edges, eliminating variations that do not correspond to real structure frontiers.

Edge detection is point-based and depends on single-pixel values. In this work, blocks—not individual pixels—are classified using segmentation , with pixel maps subsequently used for evaluation. Evaluation metrics are applied per pixel; pixels inherit the class label of their block, not their individual gray level.

Different features can be used to create masks. In [Ryan-Thiermann(2024)], image responses to structure-removal filters assign labels to pixels. These include edge micro-orientation detectors, convolution responses, or areas around pixels matched to filters sensitive to interesting or uninteresting features. Such features capture background periodicity across directions, and subtracting the filtered image from the original produces a mask used to remove large uninteresting regions.

This approach is applied in cell imaging to remove cytoplasmic noise.

The division of an image into two parts can be similar to the detection of occlusion as in [JYLee(2021)], that is, resolving what constitutes background or foreground (foreground-background separation).

Methods as in [LiShengli(2024)] use a known segmentation technique called ”Region Growing,” which progressively includes more pixels in a developing region when complex arrays of features can be attached to individual pixels. The drawback is the need to choose a seed pixel and define a growing criterion based on similarity for pixel inclusion. The advantage is that the block-based nature of image analysis is avoided, allowing more flexible growth. The referenced work employs “shifted growing,” moving the seed pixel along specific paths to ensure local growth is not interrupted by similarity gaps in one direction. Provided the background is relatively more homogeneous than the foreground , this method can robustly separate them. For example, in theory, it could be applied to Fig. 2a, where a linear background (e.g., across a horizontal stripe) is observed. However, when the background often mixes with the foreground, surrounds it, or is poorly separated, region growing segmentation can be problematic. In the present work, these background gaps are unknown a priori, and edges (area frontiers) are blurred. Hence, a blind statistical separation was chosen.

Methods from motion analysis can also be used to test the continuity across segmentation results. One can track background changes and model them. In [HWei(2018)], a motion paradigm is discussed in which changes in a local image due to motion are contrasted with changes due to spatial variation. The argument is that the spatial background model is fixed, while the background model due to motion can change. The decision involves determining when local information changes more (a) when a local block moves, or (b) when moving to the next block in a still image. The method used here considers classification parameters (example, class centers and the relative confidences of block memberships to them) across the entire image content, rather than locally. This can be more robust (inclusive of all changes) compared to the growing model. The referenced work models the still background using the ”Single Gaussian Model” and assumes a time-invariant mean and standard deviation for background blocks. The paper notes that this model is unsuitable under motion, such as varying illumination, which introduces time fluctuations. Using motion models for foreground / background separation allows adjusting a model (e.g., a Gaussian) across a few frames if the motion is smooth, and the predicted changes adapt accordingly. In the present work, no explicit model is built; the membership criterion results from KL or KS tests. Beyond the spatial consistency of KS/KL ($p-value$)s, a binary criterion based on classification map agreement on binary pixels is also used.

A combined approach for foreground / background separation detection and motion analysis in medical imaging is presented in [ChaochaoZhou(2024)]. The application addresses motion artifacts in cranial imaging and noise from slight subject motion. Digital subtraction angiography (DSA) benefits from capturing detailed vessel structures to detect abnormalities such as tumors or circulation issues, where borders must be identified with high accuracy. Small head movements during CT scans can limit accuracy, and robust vessel border modeling across 2D scans (via registration and matching) can help isolate the true background (non-vessel pixels) and detect pathological deformations. This approach uses a mask (”precontrast fluoroscopic background”) built from annotated GT blocks or healthy image blocks, isolating damaged vessel areas through a foreground / background separation process. Vessels can then be reconstructed, and issues identified. Given that vessels have tube- or fiber-like structures with roughly Gaussian cross sections, they cannot occupy less than 4 pixels per dimension.

In specialized Cryo-EM tomography work, the background can be modeled as noise [Mansournia2018SeparationIL(2018)]. Molecular parts (single particles) in 3D Cryo-EM volumes are foreground, with the rest being additive Gaussian background noise.

A similar Cryo-EM study [JoseLuisVilas(2022)] highlights that imaging artifacts, such as projection misalignment or microscope defects, complicate foreground / background separation detection. Volumetric contrast preservation can help separate foreground , while high FFT content may correspond to background and require filtering. Particle sizes vary across 3D space, unlike fixed-size blocks, contrasting with the block-based method adopted here, which classifies background or foreground blocks from variable reference training blocks.

As noted in [Xuebin(2025)], foreground / background separation can depend on image resolution, especially under high aperture conditions with limited pixel depth. Blurred regions may be assigned to foreground , sharp regions to foreground [sic]. For Fig. 2a, this approach is inapplicable, but adjusting classification parameters or block size can reveal a new background reflecting the desired object size, such as actin-villin fiber diameters or mouse brain regions. Resolution changes can help delineate fiber connections, neuronal synapse borders, and improve 3D projection matching [HamptonCM(2008)], [JeongM(2016)].

The concept of foreground / background separation can be generalized [DebaniPrasadMishra(2024)] by assigning specific clusters or classes of interest (example, faces, objects) to foreground and the rest to background. Occluded or overlapping objects can be defined as background , while clean objects are foreground. Manual cropping of faces or scene objects allows training an SVM classifier to identify foreground while keeping other areas as background.

In [SuraSabahKeiteb(2024)], Logistic Regression was used to detect faces in images. In the present work, Logistic Regression classifies foreground blocks of interest versus background blocks. Unlike the classical feature-based approach in that paper, this work relies on statistical tests for significant similarity to a few known training blocks.

II. METHODOLOGY

The paper splits the image into two main regions using clustering (as with HCL,KM methods) and very few seed points (as with LGR). Then, these these methods are tested for self-consistency using segmentation trials with different blocks-sizes or seed points. These produce segmentation maps or 3D-maps. The 3D-maps are single trial 2D-maps that are kept in 3D volumes. The block diagram of the basic processing flow can be seen in Fig. 1.

A block diagram of the basic model and the options investigated are given in Fig. 1.

In this paper, instead of explicit modeling, a range of training blocks belonging to foreground is used, with the remaining blocks classified as background. Cryo-EM research primarily focuses on 3D reconstruction from noisy 2D projections. The block-clustering approach here complements such methods by tracking foreground or background blocks across 3D scans, akin to motion analysis, allowing reconstruction of single particles from tracked blocks. Although not the focus here, it demonstrates that block-based classification can substitute exact pixel-level modeling when infeasible.

The present work investigates how different block sizes can aid in this task. A small-sized block (example 2×2 or 4×4) will likely reveal fine foreground/background transitions or repetitive patterns. Typical cases of foreground/background transitions can be observed in Fig. 2a. On the other hand, more complex cases face the challenge of difficult foreground/background separation due to high texture levels. For a tomographic image, the cytoplasm's noise in Fig. 2a can be particularly challenging.

In this context, the Hierarchical Clustering method and varying block sizes were adopted to capture a statistically meaningful separation, assuming low $p - value$ for blocks in different categories. A larger block size will "flatten" the foreground or the background , as suggested in [JYLee(2021)], and may also address overlap issues when both foreground and background pixels appear within a single block. Then, all pixels in the block are attributed to the block's category.

The minimum block size in this work is 4 pixels, making the method applicable to vessels. As a methodology, Logistic Regression classification is suitable for manually selecting a few healthy vessel pixels ($\geq$4 pixels) for training the rest of the blocks, followed by registration. A similar application is presented here for chains of protein molecules, as seen in Fig. 2a.

The first approach was to produce maps of foreground and background pixels using Logistic Regression for foreground/background separation. Image blocks of variable size ($\in 2, 4, 8, 16, 32, 64$) were classified into two basic classes. Logistic Regression requires minimal training, which can be done by selecting a few foreground or background blocks with a few mouse clicks on the image plane. A typical $(900 \times 730), \mathrm{pixels}^2$ image contains between 160 and 10,000 blocks for block sizes in the range $8, 64$. Hence, using up to 10 clicks does not constitute a fully supervised method. In this work, Logistic Regression was applied with at most 5 randomly selected annotated blocks per category (10 total for both background and foreground). KL and KS tests were then used to evaluate how well the remaining blocks (those not used as seeds) were statistically classified as foreground or background, based on hypotheses of belonging or not to either class. This approach is similar to a multiclass problem, since foreground or background classes may not be uniquely represented by a single block category.

The second method employed the HCL (Hierarchical Clustering) classification, which requires no annotated blocks but does require specifying the number of clusters to finalize the merging of block classes inherent to HCL. For a final number of clusters ($n = 2$) and variable block sizes (as above), HCL produces another map of foreground and background pixels. KL and KS tests again evaluate, in statistical terms, whether these blocks belong to the same generic cluster (foreground or background).

The third method used the standard K-Means clustering and followed the same logic: the similarity of each block to its cluster was checked, and KL/KS tests were applied to determine whether blocks could be assigned to foreground or background. It should be noted that the KS test $p{-}value$ relies on the variance of gray levels within each block category of a cluster. Therefore, the number of blocks that can be collectively assigned to a foreground or background area depends on how foreground or background areas are defined. Except for the Logistic Regression method, the other approaches—which require no manual feedback—essentially validate the use of $p - value$ as a criterion for good cluster allocation, using the remaining metrics (KM, HCL).

In the approach used here, blocks of at least 4 pixels (or $4^2 = 16$ pixels) were processed. This provided adequate smoothing without departing from the concept of classes, rather than classifying individual pixels. The goal of this work is to capture the background or foreground using the statistical discrepancy of pixel blocks, rather than relying on neighboring information. Hence, even neighboring blocks were classified independently as part of foreground or background , using information derived from distant blocks.

Classification results were obtained using Logistic Regression, Hierarchical Clustering, or K-Means over a range of block sizes. Blocks were classified, and individual pixels were assigned the class of their block to generate classification maps corresponding to the original image. The result is a binary pixel map in which pixels are labeled as 1 if there is agreement among all classification maps regarding foreground or background membership. Additionally, maps were

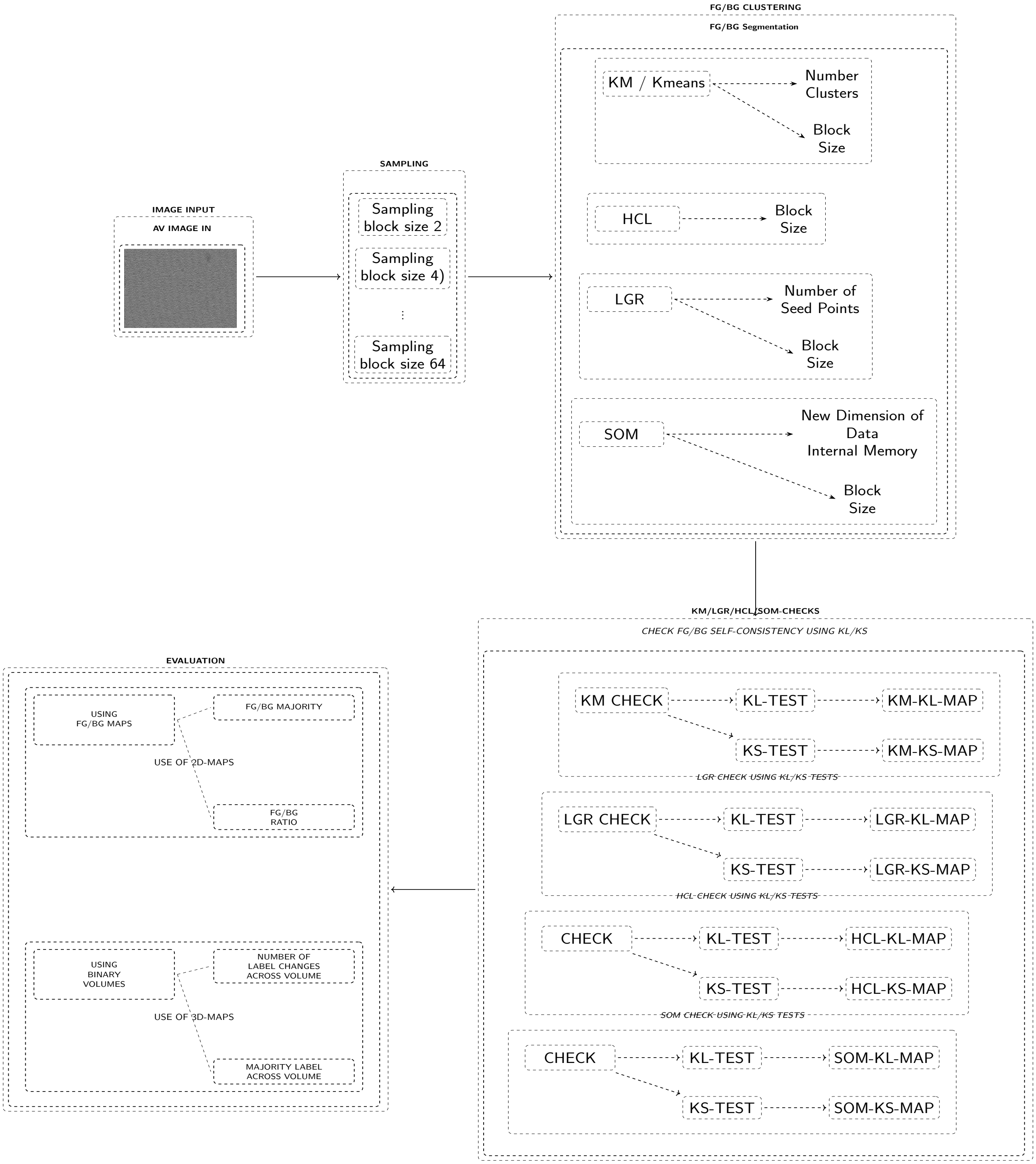
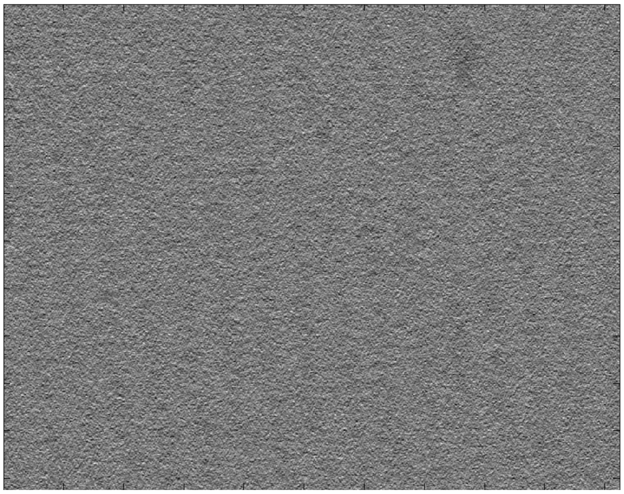


Fig. 1: The flow diagram of segmentation and testing (1)



(a) AV image

Fig. 2: The foreground and background are statistically separated in 2 different image kinds (2a) (or *'AV'*) a Cryo-EM image from a molecular environment

generated for low and high ($p{-}value$)s, not only for agreement maps. In these $p - value$ maps, a binary 1 corresponds to a high $p - value$, indicating that the corresponding blocks are classified in the same class. To compute the KS or KL maps, the typical intensity (gray-level) histograms were used for KL or KS tests to assess similarity.

To minimize bias from forcing pixels to belong to the same block-based class, different block sizes were used. For methods like Logistic Regression, which require minimal training, slightly different pixel locations were selected for the same block size. Training clicks ranged from 2 to 10 pixels for background and foreground blocks (or 1–2 per class for 5 seed pixels).

Overall, the aim was to benchmark background/foreground separation with minimal prior knowledge. This was accomplished by comparing three classification methods: (1) HCL, (2) Logistic Regression, and (3) K-Means . None of these methods uses prior models for background or foreground , except for Logistic Regression, where seed points are selected randomly from evident or less evident foreground or background areas. These include regions where foreground or background is obvious, as well as regions with low or high overlap between foreground and background.

Two types of images were analyzed. The first was a mouse brain image obtained using a Nanozoomer scanner, where foreground and background assignment can be ambiguous. The goal was not to segment the brain into functional regions but to assess classification consistency as seed point locations (Logistic Regression) or block sizes changed. The second image type was a Cryo-EM projection of molecular (protein) volumes. The foreground corresponds to protein single-particle projections that need to be tracked across successive tilts or scans as a single class. The background corresponds to the cytoplasm, which should be treated as a single class and removed.

Self-consistency was evaluated by measuring changes in classification as three major scanning parameters varied: (a) block size, (b) seed point location (when applicable), and (c) the number of clusters. Block size is critical to limiting the range of background or foreground features. Increasing block size can increase overlap between foreground and background, while smaller blocks improve classification accuracy but may produce many small foreground/background spots within larger areas.

For each block size or seed location, a new background/foreground pixel map is constructed with cluster IDs as values (0 for background, 1 for foreground). This binary map is represented as an $m \times n$ matrix $M = [0, 1]_{m,n}$ as in (1):

$$M(i,j) = \begin{cases} 1, & \text{if } M(i,j) \in FG \\ 0, & \text{if } M(i,j) \in BG \end{cases} \tag{1}$$

Such a binary matrix corresponds to a classification method $(CLS_i, i \in [1,2,3])$ for each block size $(D_i \in 2,4,8,16,32,64 = DR,\ i \in [1,6])$ or seed point. One can generate maps dependent on block size $(M_{D_k,CLS_i}(i,j))$ or a complete set of classification maps per image $(MM_{image_l.CLS_i,D_k} = M_{image_l,D_{k_1},CLS_{i_1}}, \ldots, M_{image_l,D_6,CLS_3})$.

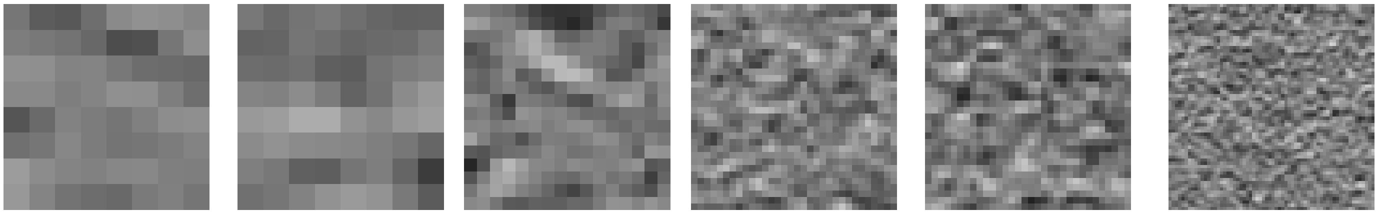
The number of clusters (for Hierarchical Clustering or K-Means) is fixed at two. Seed training point locations (for Logistic Regression) are not treated as parameters, maintaining minimal reliance on prior knowledge.

A. Use of Logistic Regression for classification

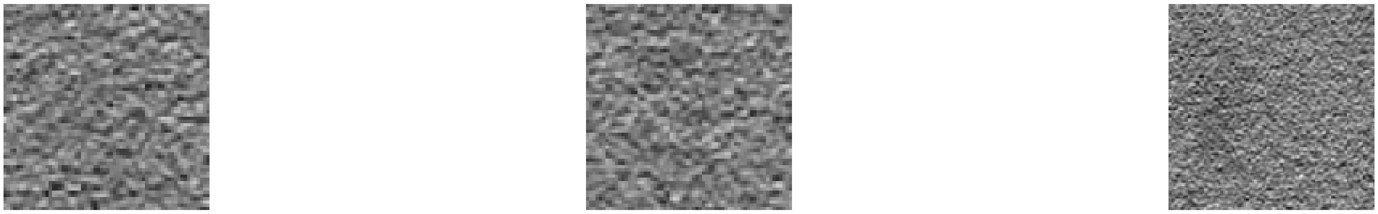
Logistic Regression is a well-known method for binary classification and is used here to classify image blocks as either FG (foreground) or BG (background). Logistic Regression is applied with minimal training. For a typical image of (728×928) pixels, which is divided into approximately 320,000 blocks of size 8×8, only up to 10 blocks are used for training across both classes. This corresponds to roughly $10/320{,}000 \times 100 = 0.003\%$ of the data. These numbers are averages and are indicative only—they do not necessarily match exactly because $N_{\text{blocks}} \times \text{BlockSize} \neq \text{ImageSize}\ (m \times n)$.

Training blocks were selected by clicking on different examples of FG or BG pixels on the image plane, keeping the total number under 10. Some indicative training blocks for the image in Fig. 2a are shown in Fig. 4a through Fig. 4e. Examples of variable-dimension blocks around seed points used with Logistic Regression are presented in Fig. 3a through Fig. 3j.

To avoid overwhelming the paper with numerous intermediate images and single-block examples, not all training points are shown. Blocks were randomly selected to cover a variety of cases, including mixed FG/BG, clean BG, and clean FG. The Logistic Regression method follows a probabilistic approach for FG/BG classification, which is a well-established standard in the literature.

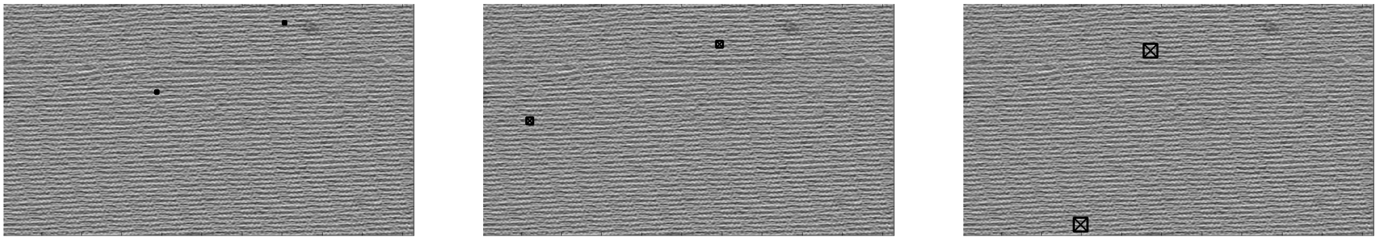


(a) AV, FG, SZ= 4 (b) AV, BG, SZ = 4 (c) AV, FG, SZ = 8 (d) AV, FG, sz= 16 (e) AV, BG, sz= 16 (f) AV, BG, SZ=16 (g) AV, FG, SZ =32



(h) AV, BG, SZ = 32 (i) AV, FG, SZ =64 (j) AV, BG, SZ= 64

Fig. 3: Indicative FG and BG training blocks for image AV (2a) at 5 block-sizes and different seed locations.



(a) AV, KM, 2 clicks, SZ = 4 ^{*1} (b) AV, KM, 2 clicks, SZ = 8 (c) AV, KM, 2 clicks, SZ= 16

(d) AV, KM, 2 clicks, SZ= 32 (e) AV, KM, 2 clicks, SZ= 64)

Fig. 4: Indicative FG and BG training blocks for image 2a in all 5 block-sizes and different seed points. The main seed if where the two diagonals cross, ^{*1} ”SZ” is the dimension of the block

$$p_{FG/BG}(i,j) = \frac{1}{1+exp^{1/B(\hat{i},j)_i^T * \hat{w}_i}}$$
$$= \begin{cases} 1 & M(i,j) \in B_(i_0,j_0)\ and\ P(i_0,j_0) \in FG \\ 0 & M(i,j) \in B_(i_0,j_0)\ and\ P(i_0,j_0) \in BG \end{cases}$$

Here, $\hat{w}_i$ is a vector of weights obtained by training the LGR classifier using pairs of the form:

$$(BLK_{tr,BG/FG}(i_{0,i_{tr}}, j_{i_{tr}})) = 1(for\ i \in [2,5], GT_{i_t r})$$

, where $(GT_{X_t r})$ is the ground-truth annotation (FG or BG) for the clicked pixel ($P_0 = (i_0, j_0)$). The term $(BLK_{i,k})$ refers to an image patch of dimension (D) centered around the chosen point:

$$B_{i_0,j_0} = [I(i,j)], for\ i \in [i_0-D/2, i_0+D/2],\ j \in [j_0-D/2, j_0+D/2]$$

with the full image denoted($I_{m\times n}$).

Thus, the Logistic Regression based method constructs a classification map using a maximum of 5 (typically 2) pre-annotated pixels, with at least one pixel per category (FG or BG). The remaining non-annotated blocks are then classified using the trained model.

B. Use of Hierarchical Clustering for classification

Then, one slightly varies the (i_0, j_0)s and gets different models and maps for 6 (D)s and 2 or 3 ((i_0, j_0))s. Then, these maps mask the original image and the resulting image blocks (B)s are statistically examined, taking their histograms that are then fed into (a) the KL divergence test, (b) the KS similarity test. The tests are conducted using as a reference histogram (the model of the test) the average histogram per image category (FG,BG). Then, for KL metrics, the KL divergence is computed as in (2), while the blocks are assigned an FG-KL value as in (3). Then, the decision is given as in (2,3,4):

$$KL : KL(H_1, H_2) = -\sum\nolimits_i H_{1_i} * log\left(\frac{H_{1i}}{H_{2i}}\right) \tag{2}$$

Let N_{FG} denote the number of FG blocks, and nB the number of intensity buckets used with the KL metric for a given block size D. The histogram of each block B_i is denoted $H(B_i) = H_i$, representing the i-th block histogram using nB buckets. Equivalently, we can write $H_i = H_i(B_i, nB)$.

We define the KL divergence between two blocks B_1 and B_2 as

$$KL(B_1, B_2) = KL(H(B_1), H(B_2))$$

The average histogram of all FG blocks is:

$$H(\hat{B_i}),\ for\ B_i \in FG$$

and similarly, the average of all BG blocks is:

$$\hat{B_{BG}} = \hat{B_i}, B_i \in BG$$

The average KL divergence of FG blocks is then:

$$KL_{FG} = \hat{KL_i}, B_i \in FG$$

denoted as *‘KlFgBlkAve‘*. For a single block B_i, the KL divergence from the FG average is:

$$KL_i = KL(BLK_i, \hat{HstFgBlkAve})$$

Finally, we define the standard deviation for both FG and BG blocks as *‘HstFgBlkStd‘* and *‘HstBgBlkStd‘*, respectively. These are used to determine the threshold for classifying a block as FG or BG, using the rule that the block's KL value should not exceed $3 \times \text{STD}_{FG \text{ or } BG}$.

$$\begin{array}{l} N_{FG} \quad is\ the\ Number\ of\ FG\ blocks \\ nB \quad is\ the\ Number\ of\ buckets \\ B_i \quad is\ the\ content\ of\ block\ i \\ H_i \quad is\ the\ histogram\ for\ B_i \\ KL(B_1, B_2) \\ \quad is\ the\ KL\ distance\ for\ blocks B_1, and B_2 \\ \hat{H_{FG}} \\ \quad is\ the\ average\ histogram\ for\ FG\ blocks \\ \hat{H_{BG}} \\ \quad is\ the\ average\ histogram\ for\ BG\ blocks \\ STD(H_{FG}) \\ \quad is\ the\ histograms\ standarddeviation\ of\ FG\ blocks \\ STD(H_{BG}) \\ \quad is\ the\ histograms\ standarddeviation\ of\ BG\ blocks \\ KL(B_i, \hat{H_{FG}}) <= 3 \times STD(H_{FG}) \\ \quad is\ the\ FG\ inclusion\ criterion \\ KL(B_i, \hat{H_{BG}}) <= 3 \times STD(H_{BG}) \\ \quad is\ the\ BG\ inclusion\ criterion \end{array} \tag{3}$$

Then, the KL evaluation is:

$$\begin{array}{l} KL_{EVAL}(i_0.j_0) \in \begin{cases} FG & Cond\ 1 \\ BG & Cond\ 2 \end{cases} \\ where: \\ Cond\ 1: \\ \quad |KL(H_i, \hat{H_{FG}}) - \hat{H_{FG}}| <= 3 * STD(KL_{FG}) \\ Cond\ 2: \\ \quad |KL(H_i, \hat{H_{BG}}) - \hat{H_{BG}}| <= 3 * STD(KL_{BG}) \end{array} \tag{4}$$

C. Use of Hierarchical Clustering for classification

The Hierarchical Clustering method is an unsupervised classification approach. For N data elements (blocks), it initially assumes a given number of clusters. This number is gradually reduced by merging similar blocks into larger clusters, while defining a distance metric that separates some blocks or splits larger clusters, depending on the hierarchical direction. The distance metric is typically Euclidean, and all blocks of the examined image are considered.

As with Logistic Regression, the analysis was conducted over the same range of block sizes (DR), but unlike Logistic Regression, the seed point location is not a parameter. Hierarchical Clustering is less biased than Logistic Regression because it does not require prior knowledge. The only prior parameter is the number of clusters sought; in this case, 2 (FG and BG). Different block sizes (D) are applied, and the resulting HCL maps are evaluated using KL and KS tests.

D. Evaluation

1) Use of KS for to value classification : Both the KL divergence test and the KS (Kolmogorov Smirnov test) test can assess the similarity of FG and BG blocks in different ways. The KS metric provides a probability that the intensity histograms (or distributions) of two blocks belong to the same class or are drawn from the same unknown underlying distribution. The $p-value$ used in the KS test represents the probability that the null hypothesis—i.e., that the two blocks are similar—holds under the χ^2 test. Here, similarity means that both blocks belong to the same class (FG or BG). The null hypothesis assumes that the intensity distributions of the two blocks both correspond to either FG or BG.

A low $p-value$ (p-value ≤ 0.05) indicates that one block is likely FG and the other BG as classified by any of the three methods (HCL, LGR, KM). Conversely, if blocks are classified in the same category (FG or BG) but KS returns a high $p-value$ (indicating dissimilarity), this signals a potential conflict between the KS test and the examined classification method, suggesting that further merging or splitting may be necessary. When the $p-value$ is low, there is agreement that both blocks belong to the same class. The relationship between KS results and a given classification method may change as the block size (D) changes.

For the KL method, one obtains a divergence map that can be converted into a probability map by assuming that divergence values for FG and BG blocks separately follow a normal distribution (N_{FG} or N_{BG}), with parameters (μ_{FG}, σ_{FG}) estimated from the sets of FG or BG blocks.

2) Evaluation by comparing clusters maps: Instead of comparing classified blocks using the KL or KS tests, one can superpose the classification maps produced over a range of block sizes or by varying the number of clusters and seed points (for LGR). As discussed, multiple label maps can be generated per image (see Section II).

Formally, let us define at least two such maps, M_1 and M_2, obtained from different trial parameters. The level of disagreement between these maps can then be quantified by checking the pixel-wise class differences for all pixels. A simple disagreement occurs when a method assigns, for a given block size and number of clusters, a label $L_1(i,j) = L_1 \rightarrow M_1$ to pixel (i,j), and for another block size or number of clusters, the label changes to $L_2(i,j) = L_2 \rightarrow M_2$.

As previously mentioned, three clustering methods were used: HCL, KM, and LGR, with 2 to 4 seed pixels (for LGR/HCL) or 6 block sizes on average. HCL performed poorly for small block sizes (2, 4) and for 2 to 4 clusters. Additional clusters (3 or 4) were tested to see whether iterative segmentation of already labeled pixels (as FG or BG) would further merge or split them into sub-FG or sub-BG areas.

Thus, for a given image, two or more label maps could be generated using the three methods and any combination of segmentation settings (block size, number of clusters, seed point locations).

Let the maps obtained from k segmentation trials be denoted as $L_k(i,j)$, for $k \geq 3$. One can then compute, pixel-wise, the number of different labels assigned at each pixel location (i,j). This indicates how many unique L_k values occurred at that pixel. The more unique labels assigned to a pixel across segmentation trials, the more changes occurred, making the pixel “vague” and a likely candidate for a border or frontier pixel between FG and BG regions.

Maps showing the number of unique labels per pixel were produced, and their histograms were computed to quantify how frequently label changes occurred. This procedure was applied for both input images, the three segmentation methods (KM, LGR, HCL), five block sizes, and 2 to 4 seed points. Overall, 3D volumes were created per image per method, where each plane of the volume represented a different sampling or segmentation setting (number of clusters, number of seed points, or block-wise/seed-wise variations). The number of unique labels was then analyzed. Due to space limitations, intermediate results are not presented. Using this criterion, the most successful method was the one in which the histogram of label uniqueness had high bins for pixels with few changes—that is, most pixels experienced few label changes, while only a few pixels underwent many changes.

The same procedure can be applied individually for each of the three segmentation methods to define two pixel categories: *‘vague‘* (likely a frontier pixel) and *‘non-vague‘* (not likely a frontier pixel). The rationale is that a pixel at the boundary between two true FG/BG areas is more likely to experience label changes across segmentation settings, whereas a pixel clearly belonging to FG or BG is likely to remain consistent.

Formally, one can construct a label-change array per pixel:

$$\overline{L_(i,j)} = \{L_1(i,j), ..., L_n(i,j)\}$$

where n is the number of different L values (not the number of segmentation trials). For $L(i,j) \in 0, 1$, one can define FG and non-FG pixels. FG/BG classification evaluation can then be based on changes in $L(i,j)$ or label differences. This is naturally formalized using 3D maps or volumes, as in (5).

$$\begin{array}{rl} V_{image_l,k}(i,j,k) = & L(i,j,k) \\ & \ni \; L(i,j,k) \in \{0,1\} \\ & \ni \; (i,j) \in [1...m, 1...n] \\ & \ni \; k \in T \in [1, NoTrials] \end{array} \tag{5}$$

The (T)s differed for each segmentation method and spanned the range of block sizes, number of clusters, and seed points. For example, HCL produced fewer (T)s because of the large volume of intermediate data generated due to its hierarchical nature, especially for small block sizes (which can produce many very small clusters).

To evaluate classification map changes across the (T)s, one measured the uniqueness of labels per pixel. As discussed, the first step was to compute the number of unique cluster labels assigned per pixel across all block sizes, seed points, or numbers of clusters. Pixels with low uniqueness values are more likely to be reliable members of FG or BG, since all classification methods tend to assign them the same label.

Formally, let $N_L(i,j)$ denote the number of distinct labels $M(i,j,k)$ for pixel (i,j) as k varies. The resulting uniqueness matrix, N_L, highlights pixels that are more likely to belong to FG or BG. Although $N_L(i,j)$ does not directly indicate FG or BG, it reflects the confidence associated with the classification: a low $N_L(i,j)$ corresponds to high confidence, while a high $N_L(i,j)$ indicates low confidence.

Several approaches could be used to evaluate how local clusters change, such as considering the impact on neighboring pixels. However, this study focuses solely on single-pixel labeling across different segmentation trials, rather than attempting to merge pixels into larger cluster segments. The metric is thus intended to quantify changes in classification rather than perform additional segmentation.

Under this map-based evaluation approach (as opposed to KL-based evaluation), the “true” ($h_{FG/BG}$) map can be approximated from the L maps using the most likely pixel labels. Consequently, a segmentation method can be assessed via the histogram of the N_L matrix, which highlights likely correct and incorrect classifications: high N_L values suggest potential misclassification, while low N_L values indicate likely correct classification across $D_i \in DR$, evaluated over the block-size range (DR), not over individual trials.

Radical changes in N_L also indicate that a method produces inconsistent classification maps and lacks stable performance. Indicative results for both stable and unstable cases are shown in Fig. 10.

This analysis can be extended to additional block sizes or numbers of clusters, creating multi-dimensional maps. In this context, $k \in [3, 10]$ for the maps $M(i,j,k)$ discussed above.

3) ROC analysis: In addition to significance tests and self-consistency measures, ROC analysis (*‘Receiver Operating Characteristic‘*) was used to further evaluate self-consistency. ROC analysis is a pixel-centered evaluation that quantifies the probability of misclassifying a pixel as foreground or background, independent of the regions to which the pixel belongs—unlike KS or KL tests, which rely on block-level or regional statistics.

The analysis relies on a so-called *'Gound Truth'*, which in this work is approximated using all intermediate classifications obtained from the trials. This choice was made because there is no established ground truth (Gound Truth) for these images, and the aim of the study is precisely to assess a minimally supervised approach. For each pixel, one can therefore consider up to N_T expected classifications (as foreground or background) across trials.

ROC curves were generated using pixel-level predictions from the Logistic Regression, K-Means , and Hierarchical Clustering two-class maps for all relevant block sizes and seed-point scenarios over N_T trials. For a pixel at location (x, y), the true positive (TP) indicator is given by $GT(x, y)$, which can be either foreground or background. Here, TP is assumed to correspond to a true foreground pixel, and TN (true negative) corresponds to a true background pixel.

The true positive rate (TPR) and false positive rate (FPR) are defined as:

$$TPR = N_{TP}/Number\ Of\ FGs\ In\ A\ Trial$$

,

$$FPR = Number\ Of\ FPs\ In\ A\ Trial/N_{BG}$$

where $N_{TP} = N_{FG}$ and $N_{TN} = N_{BG}$. TP and TN pixels are determined from the reference GT image, while FP (false positive) and FN (false negative) pixels are taken from the classification maps produced by segmentation.

Representative ROC curves are shown in Fig. 5a (case AV, KM), where "case" refers to the segmentation method or particular parameter setting used. Each plot typically contains 4–6 ROC subplots corresponding to different combinations of block size and number of clusters. For Logistic Regression, additional ROC subplots represent variations in the number of training clicks or seed points.

The AV case demonstrates superior performance: all AV ROC curves lie above the naive classifier line ($TPR = FPR$), are steeper, and achieve TPR values above 80% by the second or third trial, indicating rapid convergence of pixel classification accuracy.

The AV image has only two basic regions (filament or not).

This tends to increase the TPR slowly. Also, areas boundaries are less clear in AV images.

This also impacts the KS and KL maps previously shown in Table I, where perfect self-consistency was not observed—all values fell below 0.45. The degree of self-consistency can improve if more classes are permitted (rather than only two), allowing the method to better capture the inherent variance in the image content and produce more self-consistent classification results.

E. Comparing the KS, KL and the maps for classification evaluation

The KL divergence was used to compute the similarity between blocks produced using LGR. The similarity criterion was based on the binary notion of whether a block belonged to a FG or BG area and whether this classification was supported by the NULL hypothesis. The NULL hypothesis defined "no resemblance" when the KL divergence between two blocks—computed over their full-range histograms—differed significantly from the average KL divergence of all block pairs within the same class (FG or BG).

For all blocks classified as FG or BG by LGR, the algorithm first computed full-range histograms with 255 bins. These histograms were then used in the KL metric to generate a p-value, assuming that the KL divergences follow a normal distribution with mean equal to the average KL divergence and standard deviation equal to its observed deviation. If a block's KL p-value was below 0.05, its similarity was considered statistically insignificant, and the block was treated as not resembling the reference class. BG blocks were identified as those whose KL divergences fell in the tails of this distribution (p ¡ 0.05), ensuring that the BG area could include highly variable blocks.

Table I presents a comparative evaluation for the image shown in Fig. 2a. The table separates results by segmentation method (HCL, KM, LGR) and by evaluation metric (KS or KL). Rows correspond to different trials, reflecting various combinations of seed points and block sizes.

TABLE I: Kolmogorov Smirnov test, Kullback Leibler divergence comparisons for the images (AV) (2a)

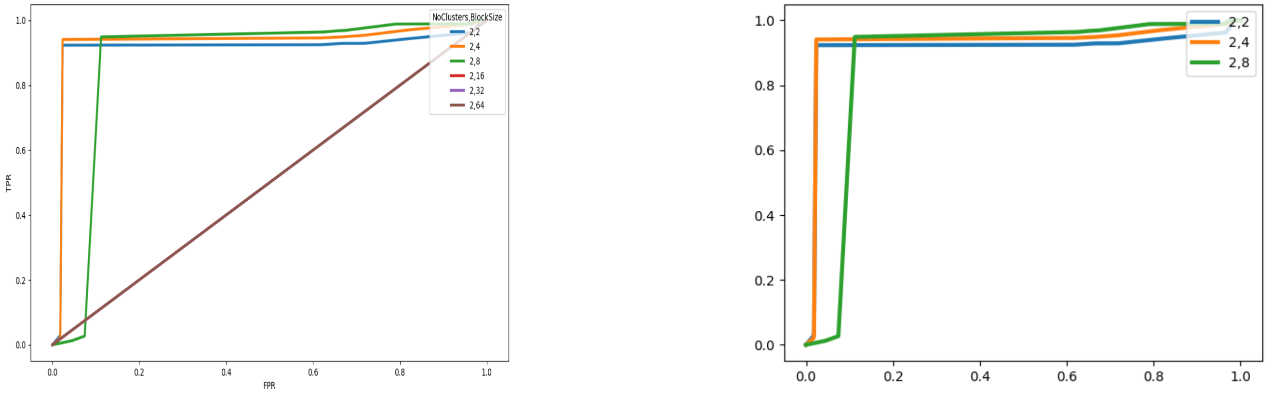
Size	Image AV					
	HCL-FG HCL-BG ^{*2}		KM-FG KM-BG		LGR-FG LGR-BG	
	KL	KS	KL	KS	KL	KS
4	0.68 0.78	0.68 0.78	0.37 0.37	0.47 0.42	0.54 0.80	0.64 0.50
8	0.96 0.95	0.75 0.69	0.96 0.95	0.65 0.59	0.43 0.86	0.45 0.49
16	0.29 0.29	0.15 0.19	0.33 0.29	0.25 0.29	0.44 0.86	0.35 0.39
32	0.30 0.32	0.29 0.21	0.30 0.32	0.22 0.21	0.37 0.90	0.39 0.21
64	0.27 0.25	0.24 0.23	0.26 0.25	0.18 0.22	0.30 0.76	0.28 0.25

^{*2} FG/BF blocks KL $p - value$

Table I shows that FG and BG blocks, evaluated using KS or KL metrics, vary roughly within the interval $[0.25, 0.45]$ for both images. The table breaks down the comparison results by dataset and covers the five block sizes discussed. Although exhaustive, the overall conclusion is that KS and KL produce roughly comparable results.

For KS, this interval represents an average confidence or p-value that is neither very high (indicating strong similarity) nor very low (indicating strong dissimilarity) for FG/BG blocks within the same segmentation. Similarly, for KL, the probability of a block falling within $3 \times STD$s from the main KL value (as in 4) also resides approximately within this interval. This suggests that KM, HCL, and LGR can all be considered reasonably self-consistent.

The limitation of considering only two clusters (FG/BG) may prevent capturing additional similarity, but some extreme cases are observed—for example, very high KL/KS values (0.68/0.78 for AV/HCL, block size 4) and very low values (0.14/0.15 for MBA/LGR). Smaller block sizes (e.g., 4) tend to favor higher similarity. Among the three methods, KM appears



(a) ROC, AV, KM

(b) ROC, AV, LGR

Fig. 5: Indicative ROC analysis plots on images MBA and AV for sample block-sizes, clusters , seed points and numbers of clicks

to yield the most self-consistent segmentation across block sizes.

F. Evaluation

G. Results using Hierarchical Clustering segmentation

The evaluation of the HCL-based foreground/background classification (Hierarchical Clustering foreround / background classification) for a variable number of final clusters in the range $2, 4, 8, 16, 32, 64$, and for block sizes $8, 16, 32, 64$, for the image in Fig. 2a, is illustrated in Fig. 9a through Fig. 9d.

One can observe that for larger block sizes and higher numbers of clusters (e.g., Fig. 9d), the KL divergence metric tends to assign lower p-values, resulting in darker KL-HCL images that indicate clearer segmentation. In contrast, smaller block sizes and fewer clusters produce finer details with more variable p-values, which are gradually consolidated into more homogeneous blocks that exhibit higher resemblance.

The multiple levels of p-values capture both high and low similarity regions: dark areas correspond to low p-values, indicating homogeneous regions, while light areas correspond to mixed regions, such as tree and sky segments toward the left of the image. Larger block sizes effectively perform image “averaging,” since more pixels are considered collectively as FG or BG.

Interestingly, varying block sizes and cluster numbers effectively mimics a spatially iterative HCL. Large FG or BG clusters obtained with a fixed 2-cluster HCL can be subdivided into smaller clusters, as if applying HCL separately on previously segmented FG or BG regions. In essence, this process further divides a known FG or BG cluster into smaller subclusters, applied only to blocks of a given FG/BG identity. This approach naturally generates a variance of p-values within initially homogeneous FG or BG areas, reflecting variable similarity across the image.

H. Results from applying KM on input images

The K-means (KM) classification was chosen as a points-free method to evaluate how BG and FG areas (BG and FG) can be separated using the simplest two-class model. Results are presented for different block sizes and either two or four clusters. The corresponding figures for the input image in Fig. 2a range from Fig. 8a to Fig. 8l.

To facilitate inspection, legends are placed below the images, and the results are also summarized using triples of the form (NoOfClusters, BlockSize, FigureID). The main triples presented are:

- (2, 8, Fig. 8c) – 2-cluster K-means segmentation with 8-block’s dimension
- (2, 4, Fig. 8b) – 2-cluster K-means segmentation with 4-block’s dimension
- (2, 2, Fig. 8a) – 2-cluster K-means segmentation with 2-block’s dimension
- (4, 64, Fig. 8l) – 4-cluster K-means segmentation with 64-block’s dimension
- (4, 32, Fig. 8e) – 4-cluster K-means segmentation with 32-block’s dimension
- (4, 16, Fig. 8d) – 4-cluster K-means segmentation with 16-block’s dimension

This notation allows a concise and systematic representation of the various segmentation results across block sizes and cluster numbers.

Comparative results using CNNs on Cryo-EM images are reported in [Adil2020], where single particles are detected using Convolutional Neural Networks (CNNs). That study achieved over 95

In contrast, the current work reached a success range using KM-KL-FG (i.e., K-means with KL applied to FG blocks) spanning $(0.26, 0.96)$, which provides an estimate of the classification error for various block sizes. This corresponds to a maximum performance of roughly 7.5

Similarly, KS values fall within the interval $(0.18, 0.47)$ and provide the cumulative probability that two blocks are not alike; higher KS values indicate that segmentation failed or is inconsistent. Table I reports average KS values for each region (FG/BG). Very low KS p-values (< 0.05) indicate that the NULL hypothesis of similarity can be rejected, meaning the blocks are statistically dissimilar.

Overall, using KS or KL tests without prior knowledge, one cannot generally reach the performance levels of CNNs (¿95%). Nevertheless, the KL maps in Fig. 9a show that some areas, particularly the dark to very dark regions (very low KL p-values), achieve strong self-consistency. Additional examples of KS consistency maps for various block sizes can be seen in Fig. 9e through Fig. 9i.

1) Results using the KL-KM method: The results obtained using the KL-KM method (that is, K-means segmentation with KL-based similarity maps) are shown in the figures from Fig. 9a to Fig. 6b, corresponding to the original input image in Fig. 2a.

To facilitate browsing the results, the figures use the same legend format as before, presented as triplets ($(NoOfClusters,\ BlockSize,\ FigureID)$).

Specifically, we have:

- (2, 2, Fig. 9a shows a 2-cluster KM segmentation with 2 pixels block's dimension) for image in Fig. 2a)
- (2, 4, Fig. 9b *'shows 2-cluster KM segmentation with 4 pixels block's dimension'*)
- (2, 16, Fig. 9c *'2-cluster KM segmentation with 16 pixels block's dimension '*)
- (2 , 64, Fig. 9d *'4-cluster KM segmentation with 64 pixels block's dimension'*)

The figures from Fig. 8a to Fig. 8l display K-means clusters that are generally clearer compared to those produced using LGR. The emergence of new structures, such as diagonal stripes, becomes visible as the number of clusters increases (for example, after Fig. 8d), similar to the patterns observed with LGR and HCL classifications. Additionally, with larger block sizes, the foreground-background separation becomes even more distinct. One could even measure the width and orientation of these stripes, if that were the goal of the application; however, such analysis is beyond the scope of the present work.

I. Results using the KL-LGR method

The method of KL resemblance using LGR classification were computed, ideally, for the defined range of block sizes, (that is, the (D_i)s), to cover an as ample as possible range of FG/BG overlaps. The term "ideally" refers to the case that the Hierarchical Clustering method that did not support the full range of (D_i)s given the nature of it (too many computations and clusters data for very small blocks). Also, the notion of very fine blocks (small) for foreground / background separation cannot always accommodate very large block sizes (for example (32)). Then, the KL match may be lost due to serious foreround / background overlap. Another parameter was the choice of the initial points, both location-wise and also as sampled from known FG or BG regions.

The results from this analysis for triplets of execution parameters of the type ($(NoOfClusters,\ BlockSize,\ FigureID)$) are:

- (2, 2, Fig. 6a) *'two (1-FG,1-BG) initial point segmentation with 4 pixels block size segmentation for image in Fig. 2a'*)
- (2, 8, Fig. 6b *'two (1-FG, 1-BG) initial point segmentation with 8 pixels block size segmentation for image in Fig. 2a'*)

The figure in Fig. 6a illustrates another interesting application of this dual foreground-background segmentation: revealing hidden sub-regions within larger initial foreground or background areas.

One unique characteristic of the original image (Fig. 2a) is that it is a texture-only image. There are no obvious foreground or background regions, except for a single particle/disc (towards the top right), which appears as the darker part of the image. This makes the image content highly dependent on fine texture variations, which are influenced by the size of the blocks used. Here, "following" means that the texture tends to be more periodic (repeating) than larger regions; thus, smaller or larger periodicities can become apparent depending on the size of the building kernel used to capture the corresponding scale.

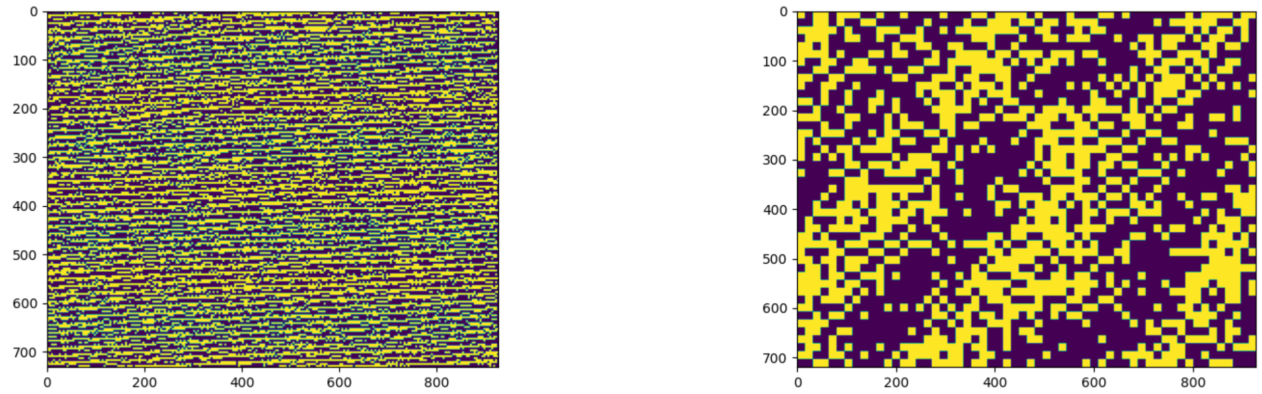
Various combinations of initial points (seed points) were selected so that a maximum of two pixels belonged to either the foreground or the background. Importantly, this approach is not supervised and does not rely on a ground truth–based classification, especially given the relatively large image size of approximately 1200×900 pixels for Fig. 2a.

The segments ' merging observed can be observed when increasing the size from 2 to 4 as in Fig. 6a.

If one compares the figures from Fig. 7a to Fig. 7c, which correspond to different methods (LGR, HCL) applied to the same input image (Fig. 2a), it is evident that the HCL method produces more homogeneous and solid regions, both for low and high numbers of clusters. One reason is that LGR is essentially a foreground–background probabilistic model that assigns local probabilities based on 2 to 4 seed pixels, as discussed earlier. Some blocks or pixels adapt well to this local model, while others do not, leading to more scattered or sparse results. In contrast, HCL employs more global, image-wide criteria to compute distances among blocks before deciding whether to merge or split groups of blocks into foreground or background regions.

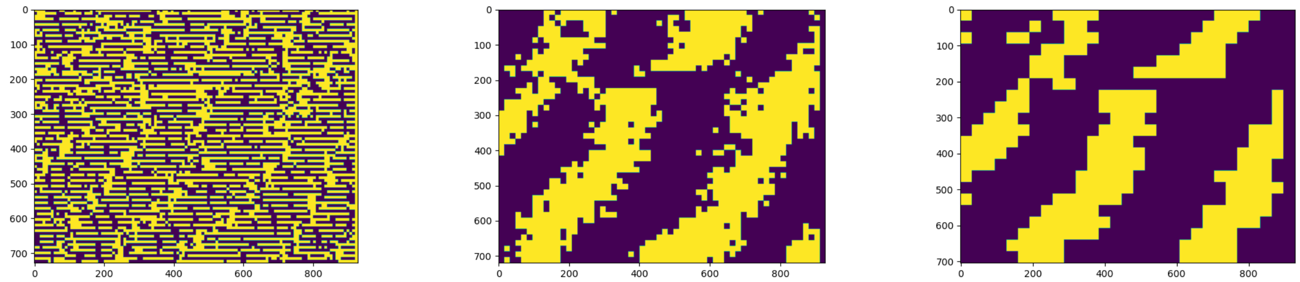
The KM–foreground/background separation method also tends to produce larger and less sparse segments. One can observe that the intrusion of foreground spots into larger foreground segments—and vice versa—is limited. This slight penetration has minimal impact and does not disrupt the homogeneous background regions (e.g., the sky area on the right). Therefore, the segmentation quality is well maintained using the adopted foreground–background approach.

Using the KL evaluation of the foreground/background segmentation of the original image (Fig. 2a), one can observe that the darker areas in the figures from Fig. 9a to Fig. 9d mostly appear at larger block sizes and when the number of HCL clusters increases only slightly (from 2 to 4). Another observation is that, as the block size increases, new structures become visible compared to the segmentation at smaller block sizes. Specifically, fine horizontal parallel stripes (actin-villin



(a) (AV, LGR, 2, SZ=2 (b) AV, LGR, 4, SZ=4

Fig. 6: Results using the LGR segmentation on image 2a for different block-sizes and LGR clusters



(a) AV, HCL, 2, SZ=8 (b) AV, HCL, 2, SZ=16 (c) AV, HCL, 2, SZ=32

Fig. 7: Results using the HCL segmentation on both images for different block-sizes and numbers of clusters , where in (x,y), (x) indicates the number of seed points and (y) the number of clusters .

fibers) are less prominent, while thicker left-diagonal stripes (running roughly from top right to bottom left) appear at approximately three to four times the thickness of the actin-villin fibers and at an orientation of roughly 45°.

The KL comparison of HCL blocks for the image in Fig. 2a is illustrated in Fig. 9c, which shows grayscale images of p-values. Darker blocks, corresponding to lower p-values and thus higher similarity, are mostly observed for small block sizes ($\leq$8) along the horizontal actin-villin fibers. As block size increases, new darker stripes—indicating more homogeneous regions—start to emerge. For very large block sizes (64), the range of p-values broadens, with both very dark and nearly white pixels, as averaging over larger blocks increases similarity (producing lower p-values).

These observations are clearly visible in the figures from Fig. 9a to Fig. 9d, corresponding to the image in Fig. 2a. To facilitate navigation, legends below the images are presented as triplets ($NoOfClusters$, $BlockSize$, $FigureID$), specifically:

- (2, 8,Fig. 9a *‘final 2-cluster HCL KL segmentation with 8 pixels block size‘*)
- (2, 32, Fig. 9b, *‘final 2-cluster Hierarchical Clustering Kullback Leibler divergence segmentation with 16 pixels block size‘*
- (2, 64, Fig. 9d *‘final 4-cluster Kullback Leibler divergence Hierarchical Clustering segmentation with 8 pixels block size‘*))

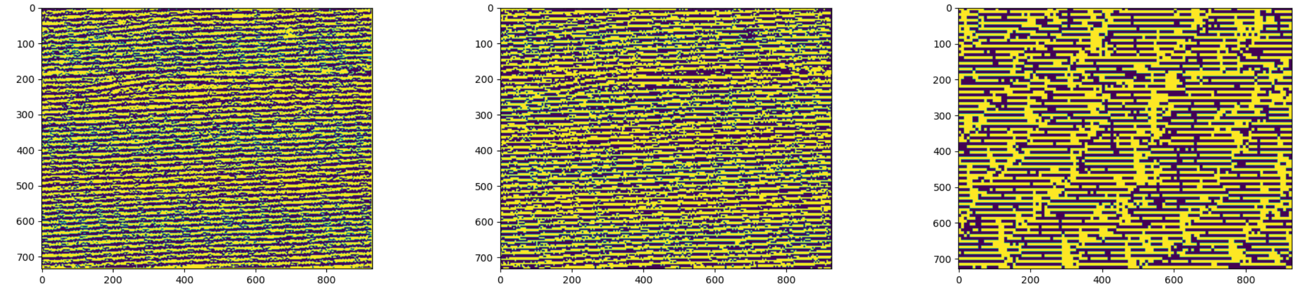
J. Evaluation of results using label changes

Evaluation maps produced as in section II-D2 can be seen in the figures from Fig. 11a when HCL is applied on Fig. 2a, on Fig. 11d when LGR is applied on Fig. 2a.

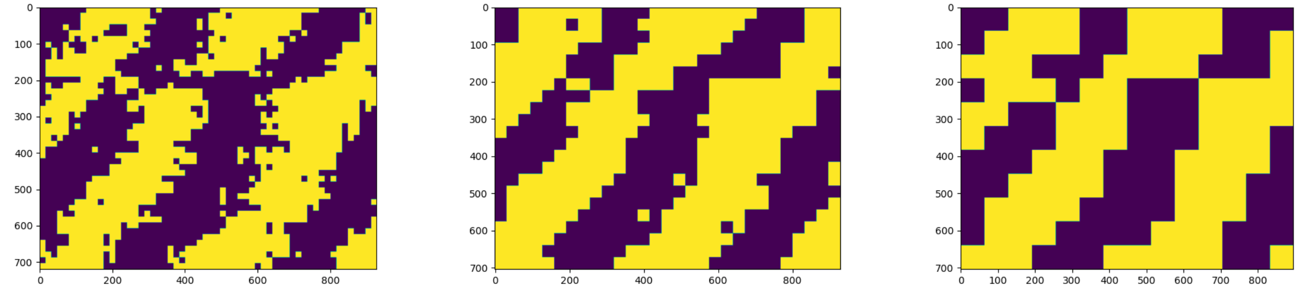
Indicative histograms as discussed in section II-D2 are given in Fig. 11c.

Relevant histograms for the label-change maps (the N_L matrices described in section II-D2), which indicate the distribution of pixels with varying labels versus stable labels, are shown in the figures from Fig. 11c for the image in Fig. 2a through Fig. 11c for the same image. These histograms highlight which pixels frequently change classification across segmentation trials and which remain consistent, providing insight into likely frontier pixels versus reliably classified foreground or background pixels.

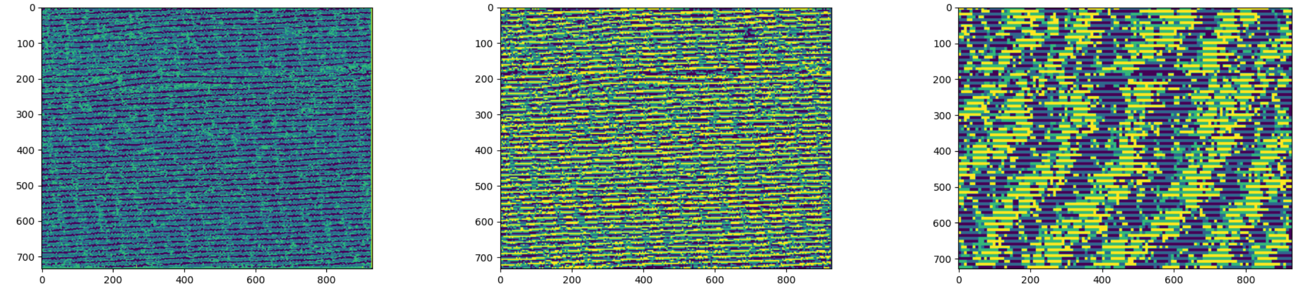
Overall, the KM classification method proved superior for foreground / background separation segmentation, taking into



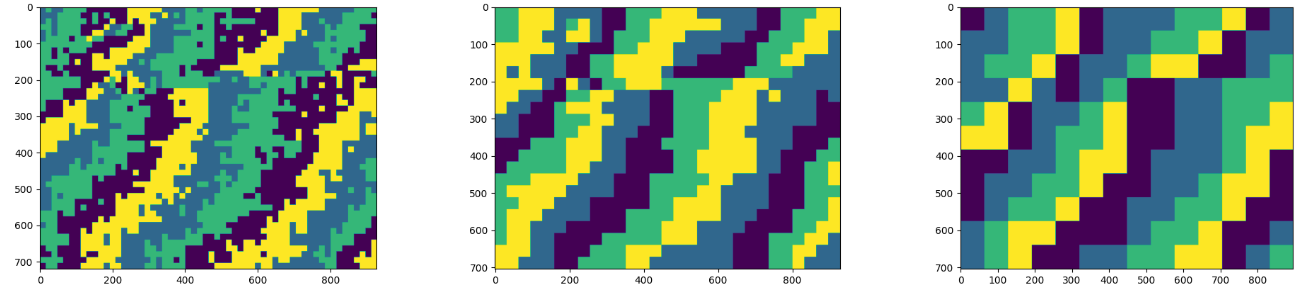
(a) AV, KM, 2,SZ=2 (b) AV, KM, 2,SZ=4 (c) AV, KM, 2,SZ=8



(d) AV, KM, 2,SZ=16 (e) AV, KM, 2,SZ=32 (f) AV, KM, 2,SZ=64



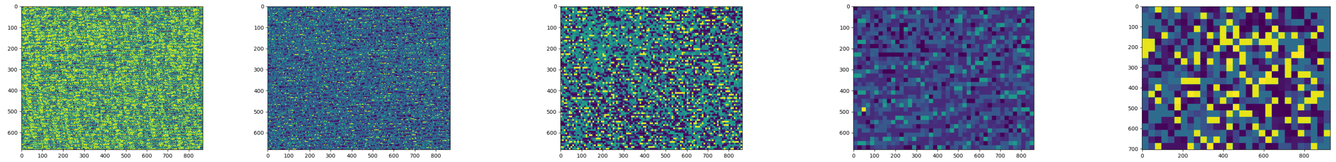
(g) AV, KM, 4,sz=2 (h) AV, KM, 4,SZ=4 (i) AV, KM, 4,SZ=8



(j) KM, 4, SZ=16 (k) AV, KM, 4,SZ=32 (l) AV, KM, 4,SZ=64

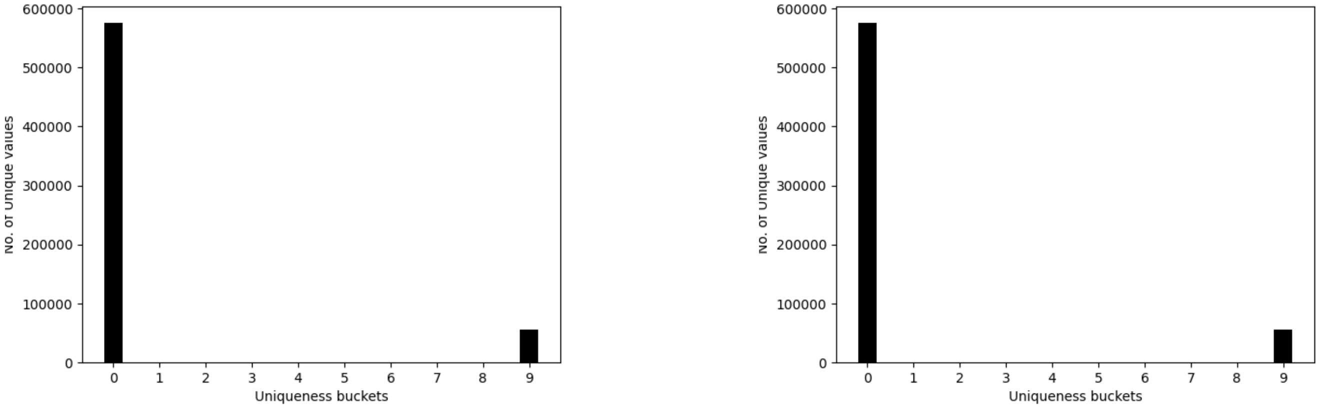
Fig. 8: Results using the KM segmentation on image 2a (AV) for different block-sizes and KM clusters , where in (x,y), (x) indicates the number of seed points and (y) the number of clusters .

(a) AV, KL, HCL, 2, SZ=8 (b) AV, KL, HCL, 2, SZ=32 (c) AV, KL, HCL, 2, SZ=16 (d) AV, KL, HCL, 2, SZ=64



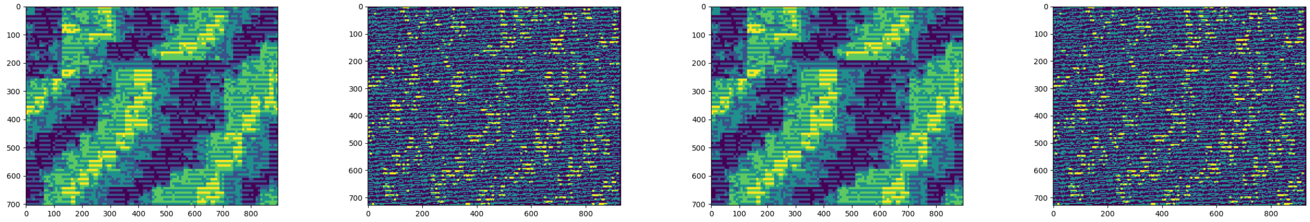
(e) AV, KS, KM, SZ=2 (f) AV, KS, KM, SZ=4 (g) AV, KS, KM, SZ=8 (h) AV, KS, KM, SZ=16 (i) AV, KS, KM, SZ=36

Fig. 9: Kolmogorov Smirnov test and Kullback Leibler divergence results using the KM segmentation on image 2a (AV) for different block-sizes or Hierarchical Clustering clusters , where (x) indicates the number of seed points and (y) in (x,y) the number of clusters .



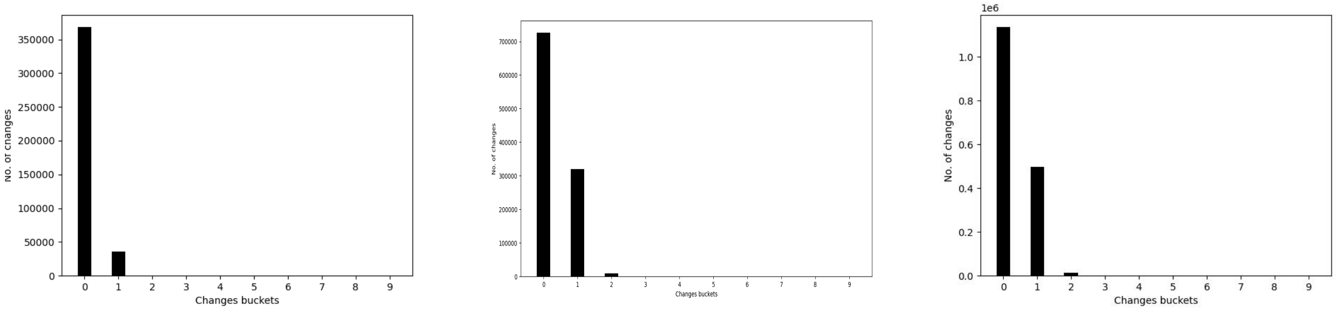
(a) AV, LGR, Label changes counts using Logistic Regression for image Fig. 2a (b) AV, LGR, Label changes counts using KM for image Fig. 2a

Fig. 10: Evaluation results using the labels change method and 3 classification methods. Large changes frequencies show low segmentation performance. The parameters are $(Number\ Of\ Changes\ (x-axis), Frequency\ OF\ Number\ Of\ Changes\ (y-axis))$



(a) AV, HCL (b) AV, HCL (c) AV, HCL (d) AV, HCL

Fig. 11: Evaluation results using the labels change map method. Light areas are more "vague"(many label changes) pixels while darker ones are known fixed label ones (either FG orBG)



(a) AV, KM, Fig.2a (b) AV, HCL,Fig.2a (c) AV, LGR LGR,Fig.2a

Fig. 12: Evaluation results using the labels change per image and classification method used

account both the standard deviation of pixel label changes and the consistency maps shown in the figures in Fig. 10.

III. ACKNOWLEDGMENTS

The author would like to thank the image provider for making the AV image open for research. This is also acknowledged in the references

REFERENCES

[Dong2018SingleLabelMI(2018)] Qi Dong and Xiatian Zhu and Shaogang Gong, *Single-Label Multi-Class Image Classification by Deep Logistic Regression*, arXiv, Vol. abs/1811.08400, Available at: https://api.semanticscholar.org/CorpusID:53786787, https://export.arxiv.org/pdf/1811.08400v1.pdf

[DebaniPrasadMishra(2024)] Debani Prasad Mishra, Sanhita Mishra, Smrutisikha Jena, Surender Reddy Salkuti, *Image Classification Using Machine Learning*, Indonesian Journal of Electrical Engineering and Computer Science, Vol. 31, No. 3, pp. 1551–1558, DOI: 10.11591/ijeecs.v31.i3.pp1551-1558

[HamptonCM(2008)] Hampton CM, Liu J, Taylor DW, DeRosier DJ, Taylor KA, *The 3D Structure of Villin as an Unusual F-Actin Crosslinker*, Structure, 2008 Dec 10; 16(12), pp. 1882–1891, doi: 10.1016/j.str.2008.09.015, PMID: 19081064, PMCID: PMC2782859

[av(2008)] Hampton CM, ResearchGate, Available at: https://www.researchgate.net/figure/Electron-Micrographs-of-F-ActinVillin-Crosslinks-A-Electron-micrograph\-of-a_fig4_23661890

[JeongM(2016)] Jeong M, Kim Y, Kim J, Ferrante DD, Mitra PP, Osten P, Kim D, *Comparative Three-Dimensional Connectome Map of Motor Cortical Projections in the Mouse Brain*, Scientific Reports, 2016 Feb 2; 6(1), Article 20072, doi: 10.1038/srep20072

[Phalempin(2021)] Phalempin M, Lippold E, Vetterlein D et al., *An Improved Method for the Segmentation of Roots from X-ray Computed Tomography 3D Images: Rootine v.2*, Plant Methods, 2021; 17(39), doi: 10.1186/s13007-021-00735-4

[RyanThiermann(2024)] Thiermann R, Sandler M, Ahirjohn G, Sauls T, Schroeder J, Brown S, Le Treut G, Si F, Dongyang L, Wang JD, Jun S, *Tools and Methods for High-Throughput Single-Cell Imaging with the Mother Machine*, eLife, 2024; 12: e88463, doi: 10.7554/eLife.88463.4, Available at: https://elifesciences.org/articles/88463

[JYLee(2021)] J. Y. Lee, R. H. Park, J. Kim, *Occlusion Handling by Successively Excluding Foregrounds for Light Field Depth Estimation Based on Foreground-Background Separation*, IEEE Access, 2021; 9, pp. 103927–103936, doi: 10.1109/ACCESS.2021.3098819, Available at: https://www.researchgate.net/publication/353390462_Occlusion_Handling_by_Successively_Excluding_Foregrounds_for_Light_Field_Depth_Estimation_Based_on_Foreground-Background_Separation

[LiShengli(2024)] Li S, Cao Z, Li J, Li P, Xu B, Gao W, *Complex Background Removal Method for Video of an Inclined Cable Based on UAV Detection of Structural Vibration Frequencies*, Structural Control and Health Monitoring, 2024; Article 1630928, 25 pages, doi: 10.1155/2024/1630928, Available at: https://onlinelibrary.wiley.com/doi/epdf/10.1155/2024/1630928

[HWei(2018)] H. Wei, Q. Peng, *A Block-Wise Frame Difference Method for Real-Time Video Motion Detection*, International Journal of Advanced Robotic Systems, 2018; 15(4), doi: 10.1177/1729881418783633, Available at: https://journals.sagepub.com/doi/10.1177/1729881418783633

[ChaochaoZhou(2024)] Zhou C, Abdalla RN, An D, Faruqui SHA, Sadrieh T, Alzein M, Nehme R, Shaibani A, Ansari SA, Cantrell DR, *Reducing Motion Artifacts in Craniocervical Background Subtraction Angiography with Deformable Registration and Unsupervised Deep Learning*, Radiology Advances, 2024; 1(3), September 2024, doi: 10.1093/radadv/umae020, Available at: https://academic.oup.com/radadv/article/1/3/umae020/7727676?login=false

[Mansournia2018SeparationIL(2018)] Mansournia MA, Geroldinger A, Greenland S, Heinze G, *Separation in Logistic Regression: Causes, Consequences, and Control*, American Journal of Epidemiology, 2018; 187(4), pp. 864–870, doi: 10.1093/aje/kwx330, Available at: https://www.frontiersin.org/journals/molecular-biosciences/articles/10.3389/fmolb.2024.1393564

[JoseLuisVilas(2022)] Vilas JL, Carazo JM, Sorzano COS, *Emerging Themes in CryoEM – Single Particle Analysis Image Processing*, Chemical Reviews, 2022; 122(17), pp. 13915–13951, doi: 10.1021/acs.chemrev.1c00850

[Xuebin(2025)] Liu X, Chen Y, Zhao C, Yang J, Deng H, *Foreground-Background Separation and Deblurring Super-Resolution Method*, Optics and Lasers in Engineering, 2025; 184, Article 108629, ISSN 0143-8166, doi: 10.1016/j.optlaseng.2024.108629, Available at: https://www.sciencedirect.com/science/article/pii/S0143816624006079

[SuraSabahKeiteb(2024)] Keiteb SS, Fadam EA, *Face Image Recognition by Using Some Dimension Reduction Algorithms and Logistic Regression*, Journal of Economics and Administrative Sciences, 2024; 30(143), pp. 405–419, doi: 10.33095/2k5d7267, Available at: https://www.researchgate.net/publication/385058088_Face_Image_Recognition_by_using_some_Dimension_Reduction_Algorithms_and_Logistic_Regression

[Adil2020] Adil M, *DeepCryoPicker: Fully Automated Deep Neural Network for Single Protein Particle Picking in Cryo-EM*, BMC Bioinformatics, 2020; 21:509, pp. 1–38, doi: 10.1186/s12859-020-03809-7