THE ACTIVE SEGMENTATION PLATFORM FOR MICROSCOPIC IMAGE CLASSIFICATION AND SEGMENTATION

SUMIT VOHRA[1], DIMITER PRODANOV[2,3]

[1] *ZIB, Berlin, Germany* [2] *NERF, Kapeldreef 75, 3001 Leuven, Belgium;* [3] *MMSDP, IICT, BAS, Sofia, Bulgaria*

Abstract. Image segmentation and classification still represent an active area of research since no universal solution can be identified. Established segmentation algorithms like thresholding are problem specific, treat well the easy cases and mostly relied on single parameter i.e intensity. Machine learning approaches offer alternatives where predefined features are combined into different classifiers. On the other hand, the outcome of machine learning is only as good as the underlying feature space. Differential geometry can substantially improve the outcome of machine learning since it can enrich the underlying feature space with new geometrical objects, called invariants. In this way, the geometrical features form a high-dimensional feature space for each pixel, where original objects can be resolved. Alternatives based on the geometry of the image scale-invariant interest points have been exploited successfully in the field of computer vision. Here, we integrate geometrical feature extraction based on signal processing, machine learning and input relying on domain knowledge. The approach is exemplified on the ISBI 2012 image segmentation challenge data set. As a second application we demonstrate powerful image classification functionality based on the same principles, which was applied to the HeLa and HEp-2 data sets. Obtained results demonstrate that feature space enrichment properly balanced with feature selection functionality can achieve performance comparable to deep learning approaches.

1. Introduction

Segmentation of cells and sub-cellular structures (e.g. organelles) is a frequent intermediate step in cell biology and pathology [30, 44]. Image segmentation can be broadly defined as isolating objects of interest from an image and gathering statistics about them by performing measurements. Automated approaches for cell segmentation have a long history, without ever reaching a universal solution [30]. This is far from a surprise, because the underlying physical modalities are very diverse. Sample preparation can be also quite different, depending on the particular experimental question – i.e. light microscopy using colored histological stains, which is still the standard in pathology; vs. fluorescent microscopy using various fluorescent stains or genetic fluorescent markers, such as the green fluorescent protein GFP; vs. transmission electron microscopy (TEM) using heavy metal contrasts

E-mail address: dimiter.prodanov@imec.be; dimiterpp@gmail.com.

(for example osmium) for sub cellular structures; vs. phase contrast microscopy for live cell imaging.

The majority of cell segmentation methods are based on only a few basic approaches. Following in part Mejiering[30] these can be classified into:

- histogram based; for example various intensity thresholding approaches [41]; mixture models, for example k-means or Gaussian
- edge-based; for example using Canny or Laplacian-of-Gaussian filters
- morphological operations; for example using morphological top-hats.
- region-based; for example using watersheds.
- deformable model fitting; for example using level-sets

Algorithms tailored for one imaging modality are very likely to fail in other modalities. For example, intensity thresholding completely fails if there is a large intensity variation of the image background, for example in a dense tissue sample. Various pre-processing steps, such as morphological top-hats of high-pass filtering can be implemented, but they are only suitable for images, which are sufficiently similar to the test batches for which the pre-processing is optimized [35].

On the other hand, the manual segmentation is labor intensive and prone to intersession and inter-observer variations. The segmentation of microscopic images is a non-trivial problem. While in the past such segmentation was performed by experts, the demands and intensity of the present experimental approaches makes such an approach non-viable. While simple protocols, such as cell counting (e.g. in a live/dead assay), can be addressed adequately by approximate segmentation, this is far from a complete morphological characterization, where the shape and orientation are to be quantified. Cells frequently have complex shapes, examples here include astrocytes, microglia, dendritic cells, etc. On the second place, in some cases, structures within some cell types can be of interest - i.e. mitochondria, synaptic vesicles, nuclei etc. As an example of the traditional approach can be identified NucleusJ [33]. It has a specific application targeting nuclear segmentation, morphometry and statistical analysis of the nucleus.

The question then arises how can the expert knowledge be harnessed by computer algorithms in the best way in order to develop versatile segmentation solutions able to address a wide range of segmentation problems. Ideally, such an approach would be

(1) deterministic, in order to ensure reproducibility of the results;
(2) providing an uncertainty measure of the segmentation outcome;
(3) able to provide some insight into how the morphology is objectively described
(4) able to learn new information whenever such is presented

There are numerous open-source programs for biomedical image processing. This can be attributed in part to the fact that in the past every imaging center developed its own analysis tools. The situation changed gradually after 2005 with the gain of popularity of ImageJ and its offshoots Fiji and ImageJ2 [37], so at present the ImageJ family of software is a *de facto* standard for biological applications [1, 40]. What made ImageJ and its constellation unique is the fact that the original design was minimalistic, focusing on the essential features, which could nevertheless be extended in almost any direction by contributed plugins, which seamlessly integrate into the main user interface. So that without emphasis on programming paradigms and particular frameworks the entry point for new users and developers was low.

Further development included Fiji [38], where one of the authors was an early contributor, as well as CellProfiler[26], Icy [13], and ImageJ2. All these platforms have varying entry thresholds for new users and developers. Therefore, to truly makes impact a new solution should be at least interoperable with some of these platforms or preferably be implemented in one of them.

At present, there are several available platforms covering parts of the above requirements – Trainable Weka Segmentation (TWS) [3] (part of Fiji), Ilastik [7], Icy [13] and CellProfiler[26]. An interesting development is the QuPath platform [5]. It targets digital pathology as a main application. The platform was specifically designed to analyze and explore whole slide imaging data consisting of image tiles. This is, of course, a different path compared to ImageJ and Ilastik , where the image is a singleton object.

Texture features have been used to classify images since 1970s [20]. In a related way, image classification can be classified into 3 main categories:

- numeric engineering method (NFE); for example using compound image transforms;
- Artificial Neural networks; for example using Convolutional Neural networks (CNN) [42].
- Transport based morphometry (TBM); for example using mathematics of optimal transport method [6]

The use of NFE is the traditional way to classify images. Based on expert knowledge or fully automated way, several morphological and polynomial features are extracted. Once the features are computed, classification algorithms, such as Support Vector Machines (SVM), random forests are then applied. The main advantage of using NFE over Neural networks are they are pretty fast. On the other hand, CNNs bypass the step of feature extraction and use several fully connected convolution layers to learn features from the source images.

The present paper focuses on two related problems: automatic classification and segmentation of microscopic images. We have employed generic approaches applicable to a wide variety of microscopic images.

The paper is organized as follows. Section 2 presents the necessary background on differential geometry, scale space theory,image moments, random forests and support vector machines. Section 3 describes the user interface and data format. Section 4 describes the used data sets and the setting of the comparison tasks. Section 5 focuses on the comparison with *ilastik*[7]. Finally, section 6 discusses our design choices and the obtained results.

2. Theory

Typically, the planar images are represented as surfaces in the three-dimensional Euclidean space $\mathbb{E}^3$, where the elevation represents the signal intensity. While good for visualization purposes, such representation can be overly restrictive. In particular, one needs not presuppose the Euclidean character of the embedding space. Mathematically, it is more fruitful to treat the image as embedded in an abstract $n+2$-dimensional vector space, which is a product of the Euclidean image plane with the vector space, provided by the point-wise image features: $I' = \mathbb{E}^2 \times V_n$. In order to fully harness the powerful theory of differential geometry, certain smoothness conditions must be imposed on the data.

2.1. **Weak image differentiation.** The fact that digital images are sampled on a discrete grid may represent some difficulty, as differentiation in the literal sense is not applicable to discrete signals. Notably, naive computations using finite differences are numerically unstable and amplify the high frequency noise. This difficulty can be overcome by applying the distribution theory[16], starting from the Leibniz identity for smooth signals I and G

$$\nabla(I \star G) = (\nabla I) \star G + I \star \nabla G, \quad I, G \subset \mathbb{E}^2$$

where ∇ represents the of the gradient given by its principal components $\nabla = (\partial/\partial x,\ \partial/\partial y)$. For the whole space if the kernel vanishes sufficiently fast at infinity it follows that $(\nabla I) \star G = -I \star \nabla G$. Therefore, even for discrete images, by extension, one can define a weak differentiation operations in distributional sense in terms of convolution with the gradient of a smooth kernel function as:

$$\nabla_G I := -I \star \nabla G \tag{1}$$

This approach can be applied also to subsequent differentiation operations, in this way enriching the feature space.

2.2. **Geometrical image features.** Consider a smooth planar image $I(\mathbf{x})$. The image intensity at a certain point in the direction $\mathbf{x} + \mathbf{r}$ can be interpolated from its local neighborhood up to second order as

$$I(\mathbf{x} + \mathbf{r}) = I(\mathbf{x}) + \mathbf{r} \cdot \nabla I + \frac{1}{2}\mathbf{r}^T \cdot \mathbb{H}(I) \cdot \mathbf{r} + \mathcal{O}(\mathbf{r} \cdot \mathbf{r})$$

where the dot represents matrix multiplication. Note that to ease notation the bold font for vectors will be skipped further. The Hessian tensor is represented by the matrix of second order partial derivatives

$$\mathbb{H}(u) = \begin{pmatrix} u_{xx} & u_{xy} \\ u_{yx} & u_{yy} \end{pmatrix}$$

where sub-scripted letters are used to identify partial derivatives with respect to the coordinates. Note that partial differentiation along a certain coordinate is denoted by a sub-script. It is also assumed that the Hessian is symmetric, since for smooth signals the partial derivatives commute: $u_{xy} = u_{yx}$.

This picture is a part of the image *jet-space*, which is a higher dimensional differential vector space, as a natural basis for encoding the geometry of an image point local neighborhood [16, 27].

2.3. **Differential invariants.** There are several types of geometric features, which are useful for segmentation applications. Typical interesting image features are blobs, filaments and corners. Notably, object boundaries can be represented in terms of edges, which can be approximated by steps in image intensity. All these features can be computed from the local differential structure of the image and notably from the differential invariants. The theory will be exemplified with the Gaussian derivatives, which in view of the duality property of Eq. 1, can be used to compute the image derivatives. The first four differential invariants are given in Table 1.

The eigenvalues of the Hessian tensor are solutions of the characteristic equation

$$det\,(\mathbb{H} - \lambda \mathbb{I}) = 0,$$

First order invariants	
Gradient amplitude	$A = \sqrt{G_x^2 + G_y^2}$
Gradient orientation	$\sin\phi = G_y/\sqrt{G_x^2 + G_y^2}$
	$\cos\phi = G_x/\sqrt{G_x^2 + G_y^2}$
Second order invariants	
Laplacian	$\Delta_G = Tr\,\mathbb{H} = G_{xx} + G_{yy}$
determinant of the Hessian	$det\,\mathbb{H} = G_{xx}G_{yy} - G_{xy}^2$

TABLE 1. Low-order differential invariants

line curvature	$\left(G_{xx}G_y^2 - 2G_xG_yG_{xy} + G_x^2G_{yy}\right)/\left(G_x^2 + G_y^2\right)^{3/2}$
mean curvature	$1/2\left(\left(1 + G_x^2\right)G_{yy} - 2G_xG_yG_{xy} + \left(1 + G_y^2\right)G_{xx}\right)/\left(1 + G_x^2 + G_y^2\right)^{3/2}$
Gaussian curvature	$\left(G_{xx}G_{yy} - G_{xy}^2\right)/\left(1 + G_x^2 + G_y^2\right)^2$

TABLE 2. Curvature invariants

where $\mathbb{I}$ is the identity matrix. This is a square equation with two real roots $\lambda_{1,2}$, such that $\lambda_1 + \lambda_2 = \Delta_G$ and $\lambda_1\lambda_2 = det\,\mathbb{H}$. In addition, by the symmetry of the Hessian, the eignevectors form 2 mutually orthogonal vector fields.

If both eigenvalues are negative, this is an indication for a bright blob-like feature around the point of reference. In a similar way, if both eigenvalues are positive, there is a dark blob-like feature around the point of reference. If the eigenvalues have opposite signs this is an indication of a saddle point at this point. Therefore, the zero-crossing of the Laplacian operator can be used to delimit regions, encompassing blobs. The zero-crossings form the so-called *zero space*. This can be applied also to the normal component of the ALoG and can be used to segment blobs [34].

The number of differential invariants increases with the increase of the image dimensions, but the theory can be extended along similar lines.

2.4. **Curvature invariants.** Second-order properties of smooth surfaces are described well by two invariants of interest – mean curvature and Gaussian curvature, while of inines - by one invariant – the line curvature. Geometrically, the mean curvature is given by the divergence of the unit normal vector. Geometrically, the Gaussian curvature is the limit ratio of the solid angle subtended by the normal projection of a small surface patch divided by the area of the patch. The formulas are presented in the following Table 2. The curvature invariants are implemented as two filters – Curvatures 2D and Curvatures 3D (see Table 3).

2.5. **Structure tensor.** The structure tensor (ST) is a more abstract representation of the gradient. The tensor encodes the predominant directions of the gradient in a specified neighborhood of a point, and the degree to which those directions are coherent as a function of scale. Suppose that we have a scale-space representation of the gradient vector $\nabla_G I$. Then the structure tensor is the smoothed tensor product of the smoothed gradient vector[9]:

$$S_r(I) := G_r \star \{\nabla_G I\, \nabla_G I^T\} \tag{2}$$

From this expression it is apparent that the operator introduces smoothing on two scales. However, because of its quadratic characters the scales do not compose. $S_r(I)$ can be represented by a 2x2 matrix. Besides the information on orientation and magnitude of structures, which is already present in the gradient, the structure tensor contains some further information. This additional information has been obtained by the second smoothing step and measures the homogeneity of orientations within the neighbourhood of a pixel. The information contained in the tensor is encoded by its eigenvalues and the eignevectors. Suppose that there is no second smoothing step. Then, the maximal eigenvalue encodes the eigenvector in the direction of the gradient, while by symmetry the second eigenvector is orthogonal and is in the direction of the isophote. So that the encoded information is equivalent to the gradient. Moreover, if the second eigenvalue is 0 it means that the image has constant intensity along the direction of the second eignevector. The second smoothing step induces a weighted averaging in a larger window. A useful measure of the homogeneity of the image is the coherence

$$c = \left(\frac{\lambda_1 - \lambda_2}{\lambda_1 + \lambda_2} \right)^2$$

This quantity is 1 when the gradient is totally aligned, and 0 when it has no preferred direction. The structure tensor is implemented as the filter Gaussian Structure (see Table 3).

2.6. **Scale space theory.** In the discrete domain, smoothing leads to loss of resolution and, therefore, of some information. However, the information loss can be limited if one uses multiple smoothing scales. A systematic way to treat such information loss is offered by the scale space theory [23, 29]. The axiomatic linear scale space theory was formulated in series of works by Witkin and Koenderink [46, 25]. In its original version, the theory depends on several properties of the Gaussian filters as solutions of the diffusion equation in the scale-space generated by the image. That is, the generic smoothing kernel G is identified with a radially-symmetric Gaussian kernel of scale $s = \sigma^2$

$$G = \frac{e^{-r^2/2s}}{2\pi s}$$

Gaussian kernels provide several advantages: (i) they are rotationally invariant (ii) they do not produce artificial extrema in the resulting image (iii) successive convolutions with different kernels can be combined. Mathematically, this imposes a very useful semi-group structure, equivalent to the heat/diffusion equation. In this sense, the image structures diffuse or "melt-down", so that the rate of this diffusion indicates the "robustness" of the structure.

In its typical presentation, the scale space theory applies only smoothing steps. Later, the theory was extended to include also differentiation and thus account for the differential structure of the images [28].

Linear diffusion scale-spaces are well-posed and have a solid axiomatic foundation. On the other hand, for some applications, they have the undesirable property that they do not permit contrast enhancement and that they may blur and delocalize structures. Non-linear scale spaces try to overcome some of these limitations. Such scale spaces arise in non-linear partial-differential equation framework. The formal properties of some types of scale spaces have been established by Alvarez et al. [2] and encompass anisotropic diffusion, among other approaches.

In particular, the Laplacian operator can be decomposed into 2 mutually orthogonal components: one along the direction of the gradient and another one along the direction of the isophote [45, ch. 1]. The representation is provided below for reference:

$$\Delta_G = G_{xy} + G_{yy} = \Delta_{||G} + \Delta_{\perp G} \tag{3}$$

$$\left(G_x^2 + G_y^2\right)\Delta_{\perp G} = \left(G_x^2\right)G_{xx} + \left(2G_xG_y\right)G_{xy} + \left(G_y^2\right)G_{yy} \tag{4}$$

$$\left(G_x^2 + G_y^2\right)\Delta_{||G} = \left(G_x^2\right)G_{xx} - \left(2G_xG_y\right)G_{xy} + \left(G_y^2\right)G_{yy} \tag{5}$$

Here G is a shorthand of a convolution with a Gaussian kernel or its derivative of an appropriate order. This is implemented as the ALoG filter (see Table 3).

In a similar way, the Laplacian operator can be iterated giving rise to the Power-of-Laplacian (PoL) operator [34]:

$$\Delta_G^n I = (\underbrace{\Delta_G \ldots}_{n \text{ times}})I \tag{6}$$

This operator enhances high-frequency features of an images given the scale cut-off. This can be seen easily from the frequency response of the LoG filter $\mathcal{F}_k\Delta_G = -k^2s^2$ where k is the wave vector. The PoL operators is immediately generalizable from its Fourier space representation to the positive real domain of the exponent. This can be achieved in terms of the α-scale space theory [14], which offers interesting possibilities of a broader class of smoothing filters, which are derivable from the Riesz Laplacian operator. The PoL operators are implemented by 2 filters – BoG and LOGN for speed reasons (see Table 3).

2.7. **Image Moments.** In the continuous approximation, the moments of a function are computed by the integral

$$M_{m,n} = \int\int_I P_{m,n}(x,y)f(x,y)dxdy$$

where $P_{m,n}(x,y)$ is polynomial, parameterized by the integers m and n. For example, the raw image moments are given by the homogeneous form $P_{m,n}(x,y) = x^m y^n$. The moments, can be referred to the center of the image frame or to the center of mass of the image (x_c, y_c), in which case, $P_{m,n}(x,y) = (x-x_c)^m(y-y_c)^n$. The two main problems with such a choice is that the moments contain redundant information because the homogeneous polynomials are not orthogonal; also loss of numerical precision due to cancellation of large terms. Mathematically, a better choice of polynomial is a polynomial from an orthogonal family. Such polynomials enjoy an expansion property, that is

$$f(x,y) = \sum_{m=0}^{\infty}\sum_{n=0}^{\infty} M_{m,n}P_{m,n}(x,y)$$

Useful examples of such orthogonal families are the Legendre and Zernike polynomials. The Legendre polynomials form an orthogonal set on the interval $[-1,1]$ and are given by the equation

$$L_n(x) = \frac{1}{2^n n!}\left(\frac{d}{dx}\right)^n (x^2-1)^n$$

which gives the useful two term recurrence relation[1]

$$(n+1)L_{n+1}(x) = (2n+1)xL_n(x) - nL_{n-1}(x), \quad L_0(x) = 1,\ L_1(x) = x$$

that can be used for computation purposes.

The Zernike polynomials are normalized on the unit disk in the complex plane. They can be defined for $n - m$ even as:

$$R_n^m(r) = \sum_{l=0}^{(n-m)/2} \frac{(-)^l (n-l)!}{l!((n+m)/2-l)!((n-m)/2-l)!} r^{n-2l}$$

and 0 otherwise.

2.8. **Random Forest Classifier.** The random forests are an ensemble learning method for classification, which consists of a large number of individual decision trees [21]. For classification tasks, the output of the random forest is the class selected by most trees. It is one of the most used algorithms, because of its simplicity and diversity (it can be used for both classification and regression tasks). The random forest selects features at random and it searches for the best feature among this random sample of features. The random forest classifier is preferred over other nonlinear classifiers due to a fewer number of hyper parameters – i.e number of trees for segmentation and number of features for splitting. By default, for a segmentation problem with k features, 100 trees are used to learn the model and $\sqrt{k}$ are used for feature splitting.

2.9. **Support Vector Machines.** The Support Vector Machines (SVM) is a supervised machine learning algorithm, which is a generalization of the one-dimensional perceptron [12]. The algorithm is used to classify incoming data by calculating a hyperplane, maximizing the distance from the nearest data point on each side. If such a hyperplane exists, it is denoted as the maximum-margin hyperplane.

Several implementations of SVMs are available i.e. SMO [32], LibLinear [15] and LibSVM [11]. We have used SMO [32] because it is fast and divides the large quadratic optimization problem into sub problems and uses heuristic to solve each sub-problem. In order to validate the obtained results, we have tested all the three implementations on HeLa's data set and the results obtained are similar.

3. User interaction and Architecture

3.1. **Architecture.** The *Active Segmentation/ImageJ* (AS/IJ) platform is implemented as a plugin for the public domain image analysis platform *ImageJ*. AS/IJ is fully integrated with ImageJ. As can be seen from the theoretical overview an image filter transforms an image point to a point in a derivative image, therefore in mathematical terms it is a 1:1 transformation. On the other hand, an image moment transforms an entire image to a point, therefore it is N:1. This distinction imposes some differences in the programming architecture. The filter implement the *IFilter* interface, while the surjective transforms implement the *IMoment* interface. The filtering functionality of the platform is extendable via plugins, which are loaded automatically upon starting of the platform from the plugin path. The built-in filters are listed in Table 3. Every filter is a fully functional ImageJ plugin, which can be also used in a standalone mode.

[1] `https://mathworld.wolfram.com/LegendrePolynomial.html`

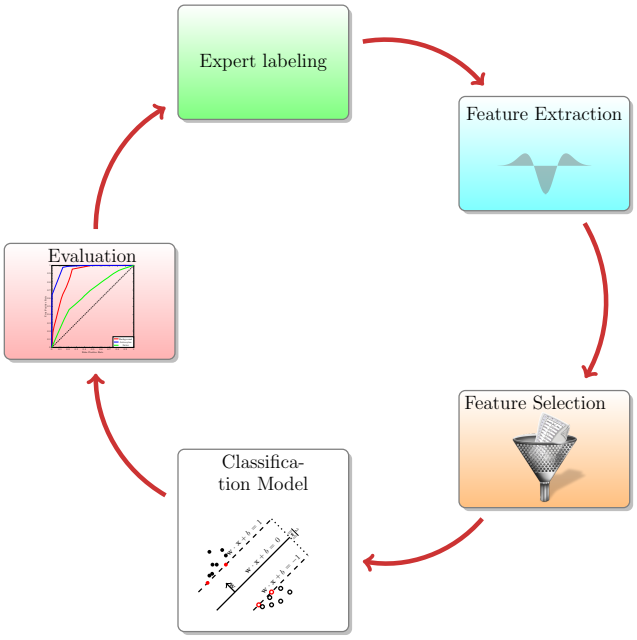

FIGURE 1. Learning cycle

We have exploited ImageJ's modularity of design and even went one step further – the filters of the AS/ImageJ can be assigned to a particular analysis project so the system would not compute the filter responses of the entire filter family. This is why for example the gradient is computed both in ALoG and the Gaussian Structure filter, for example. The signal processing approach demonstrated in the paper is based on the scale space theory [23], as a systematic way of treating image features based on differential invariants [17]. However, the software implementation is not limited by this approach and can handle any point-to-point image transform.

3.2. **Operation of the platform.** There are 2 user interaction modes: either for image segmentation or for image classification. The users can initiate analysis project or interact with existing one. If the users load an existing project they can 1) select suitable filters, 2) define classes and present instances for learning 3) select machine learning functionality; and finally 5) train and visualize the achieved segmentation. This operation mode and the associated learning cycle is schematically presented in Fig. 1. The user interface of the platform is displayed in Fig. 2.

The image classification interaction mode is similar. The texture computing functionality is enabled by default for a classification project. The built in texture features are listed in Table 4. The functionality is also extendable by a different plugin interface. The feature-pluings are also loaded automatically from the plugin path.

Both segmentation and classification results can be fine-tuned by presenting additional ROIs or by refining and editing the existing ROIs. This allows a high degree of tunability of the outcome, which corresponds with practice. This feature

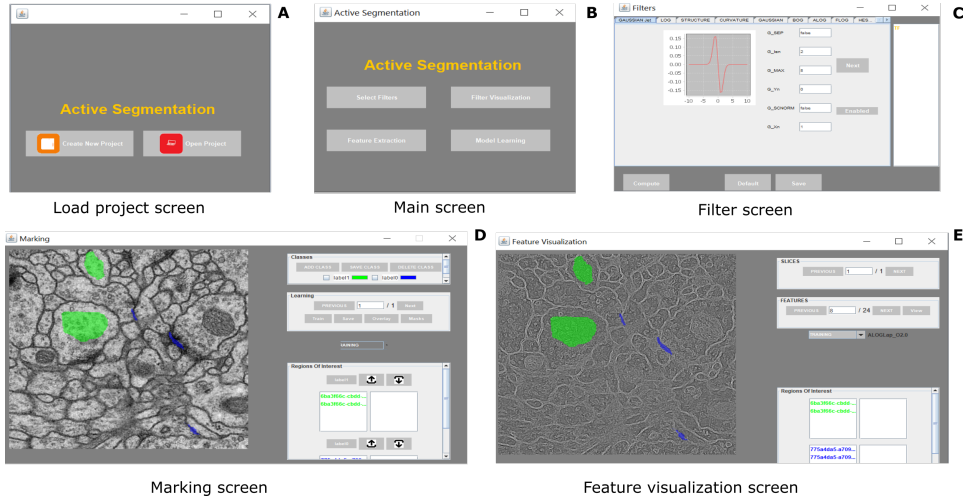

FIGURE 2. Segmentation workflow of AS/IJ

Filter	Function
LoG	Laplacian of Gaussian (LoG)
ALoG	Anisotropic decomposition of LoG, eqs. 4, 5
BoG	Bi-Laplacian of Gaussian, eq. 6
LoGN	N-th order PoL, eq. 6
Hessian	eigenvalues of the Hessian
Gaussian Jet	Gaussian Jet of order n
Gaussian Structure	Structure tensor, eq. 2
Curvatures 2D	Line curvature + Hessian determinant
Curvatures 3D	Mean + Gauss curvature of surfaces

TABLE 3. Filters implementing geometric features

Regional Feature	Function
Legendre	Legendre moments
Zernike	Zernike moments
ImageJ	ImageJ statistics
Haralick	Haralick features[20]

TABLE 4. ImageJ plugins used for computation of texture features

has been borrowed from the TWS platform, which served as an inspiration for our work.

3.3. **Metadata.** The platform produces a large number of intermediate files, which are stored locally in a predetermined folder structure. This is described in the project metadata. The project metadata are stored in JSON form, which allows also for human readability. The file stored all filter settings, Regions of Interest (ROIs), and class labels allowing for reproducibility of the obtained results. A sample project file is available as a supplementary material.

FIGURE 3. Project structure

The AS/IJ project structure is exhibited in Fig. 3. The "images" folder contains a copy of the training raw data (i.e. an image stack). The "features" folder contains all filter responses for the training data sets. The "learning" folder contains the Weka machine learning file in (Attribute-Relation File Format) ARFF format. The "tesfilters" folder contains all filter responses for the testing data sets. The "testimages" folder contains a copy of the testing raw data.

The metadata structure ensures reproducibility of the classification and segmentation results across sessions. This design choice is made specifically to ensure support of the Findability, Accessibility, Interoperability, and Reusability (FAIR) data management principles. In particular, we implement on the R-part of the principle in the way that all filtering settings are stored in the project file. Furthermore, the file format is transparent for both human eyes and algorithms as JSON is a subset of JavaScript, which is a standard web computing language.

4. Materials and Methods

A brief remark on terminology is order. The term feature, which is generally accepted in classification and machine learning literature for the purposes of the present discourse is ambiguous. As noted above, an image filter is 1:1 (i.e. injective), therefore the generated features will be denoted as *point-features* or *pixel-features*, while features generated by a surjective operation (e.g. moment computation or similar) will be denoted as *regional features*, since they are derived from a ROI. For example, the standard ImageJ regional features are of two types: i) geometrical, characterizing the primary ROI – area, bounding rectangle, centroid, perimeter, ellipse, Ferret's diameter, shape descriptors (form factor, roundness, compactness, aspect ratio); and ii) statistical, which are derived from the ROI histogram – mean, standard deviation, mode, min, max, center of mass, integrated density, median, skewness, kurtosis, and area fraction. In the subsequent discourse, the term geometrical will be interpreted only as referring to differential geometric properties.

4.1. **Data sets.**

4.1.1. *Synthetic data sets.* Two synthetic data sets were used for demonstration and debugging purposes. Both data sets contain 5 images of size 256 x 256 pixels. One was used for the segmentation task and the other – for the classification task. The first data set consists of pairs of half ellipses, having the appearance of kidney beans, superimposed on a background with or without noise. The ellipses are placed in such a way as to only minimally overlap.

For the classification task data sets, circles of random sizes were placed randomly in the image plane, while the triangles were affinely transformed and also placed randomly. The motivation behind using these two classes of figures is that Zernike moments are more sensitive toward circles while Legendre moments are more sensitive toward lines. Figure 4 (A) and (C) show representative images with and without background Gaussian noise.

4.1.2. *EM ISBI challenge data set.* Cardona et al. [10] published this data set for analysis of neuronal structures of the Drosophila ventral nerve cord. The raw data come from serial section Transmission Electron Microscopy (ssTEM). The data sets

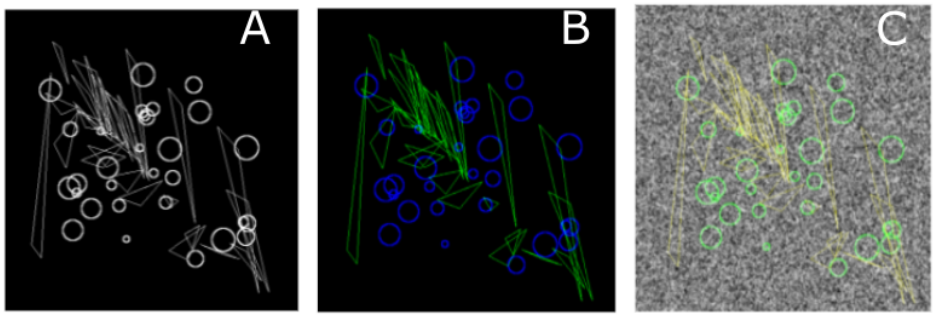

FIGURE 4. Circles/Triangles classification. A: An image generated with variable size triangles and circles on black background; B: Individual objects are classified as triangles (green) and circles (blue) on noiseless background; C: Individual objects are classified as triangles (green) and circles (blue) on noisy background.

is annotated with the segmentation ground truth as binary labels. The ground truth consists of 2 classes as shown in Fig. 2 where the black color represents membranous structures and white represents other objects. This training data set contains 30 images of 512 x 512 pixels and measures 2 x 2 x 1.5 microns, with a resolution of 4x4x50 nm/pixel. The dataset was included in the ISBI EM segmentation challenge: `https://imagej.net/events/isbi-2012-segmentation-challenge`. The images are representative of actual EM images and contain some noise and small image alignment errors. None of these problems led to any difficulties in the manual labeling of each element in the image stack by an expert human neuroanatomist. The aim of the challenge was to compare and rank the different competing methods based on their pixel and object classification accuracy. Besides the data set, the challenge defined evaluation metrics [24] based on membrane certainty and uncertainty in order to evaluate the performance of different classifiers.

4.1.3. *HeLa data set.* Murphy [31] and Boland et al.[8] have published the HeLa data set and investigated the role of expressed protein at subcellular level. In total, the data set contains 10 different classes of proteins – endoplasmic reticulum protein (ER); the Golgi proteins giantin and GPP130; the lysosomal protein LAMP2; nucleolar protein nucleolin; a mitochondrial outer membrane protein; the cytoskeletal proteins actin and tubulin; and the endosomal transferrin receptor. Figure 5 shows representative examples of each protein structure. This data set consists of 862 images of size 382 x 382 annotated with ground truth labels. The data set is imbalanced with a smaller number of samples for the mitochondria.

4.1.4. *HEp-2 data set.* The Human epithelial type-2 cells originally imaged by Sillivian Nicoliades pathology laboratory, Australia [22]. The data set was downloaded from Qi et al.[36] web-page where the authors also demonstrate an approach to classify the cells. The data set consists of 5 classes – centromere (CE), Golgi (GO), Homogeneous (HO), Nucleolar (NU), and nuclear membrane (ANC) and speckled (SP). Figure 6 shows representative images of each cell class. This data set contains 63 445 images of variable size annotated with ground truth labels, which are stored

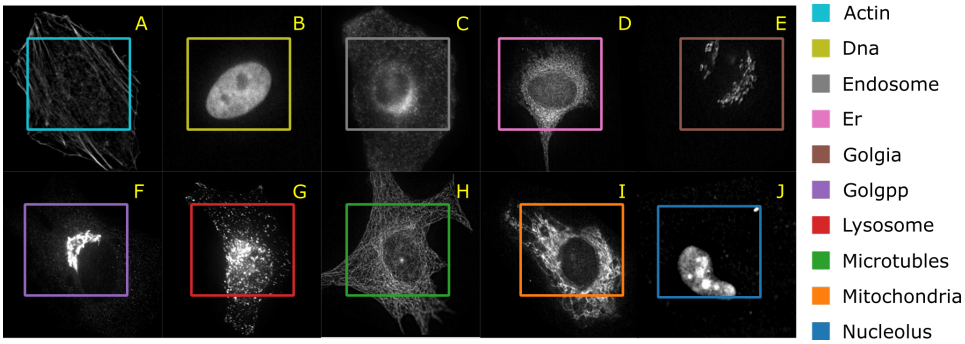

FIGURE 5. A representative image of each protein class and the average number of correctly classified instances in 10 cross validation A: filamentous actin labeled with rhodamine-phalloidin (Actin); B: DNA labeled with DAPI (DNA); C: endoplasmic reticulum protein (ER); D: transferrin receptor (endosomes); E: Golgi protein giantin (Golgi); F: Golgi protein GPP130 (Golgpp); G: lysosomal protein LAMP2 (lysosomes); H: mitochondrial protein (mitochondria); I: nucleolar protein nucleolin (Nucleolus)

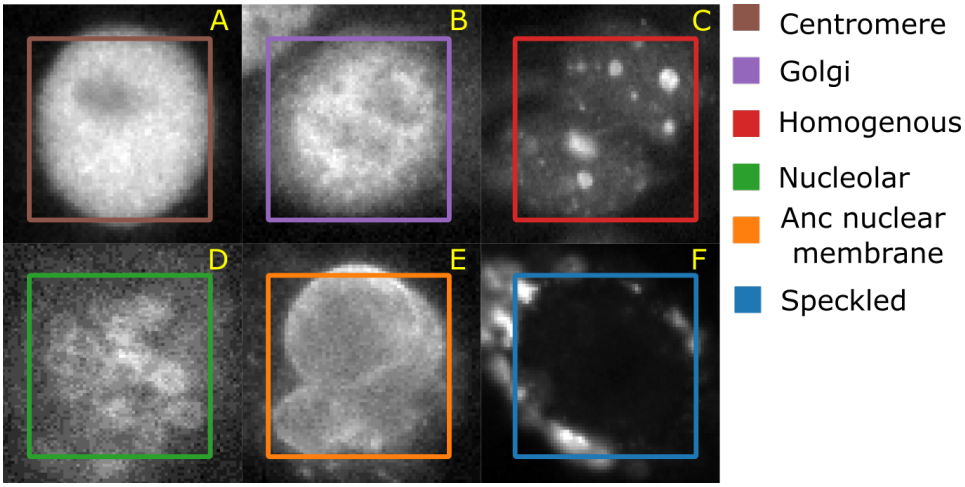

FIGURE 6. A representative image of each cell class and the average number of correctly classified instances in 10 cross validation. A: Centromere (CT); B: Golgi apparatus (GG); C:Homogeneous (HM); D: Nucleolar (NCL); E: Anc nuclear membrane (ANM); F: Speckled (SP)

in a separate Matlab file. It should be noted that both HeLa and HEp-2 data sets are imbalanced.

4.2. **Image Segmentation Task.** The typical use of the segmentation functionality consists of 4 steps as follows: the first step is to provide defined samples of each class in different colors e.g. nucleous, background etc. as shown in Fig. 2. The

samples can be outlined using rectangular, polygon or composite regions of interest ROIs in ImageJ. Secondly, an array of geometric features at different scales are computed using the filters listed in Table 3 and a feature vector is generated for each pixel. In the third step, most discriminative features are selected using some useful heuristic – i.e. for example by correlation-based feature selection (CFS) [19], which uses greedy hill climbing and best first search to explore the feature space. Various other feature selection techniques i.e. Principal Component Analysis (PCA), rank-based methods or others could also be used. Next, the vectors are classified. For the purposes of the paper, in order to allow for comparisons with other platforms, the random forest classifier[18] with 100 default trees was used to train the model. The reasons behind using random forest as the default classifier for segmentation are: 1) it allowed us to compare our results with Ilastik[7]; 2) it has a smaller number of parameters and scales matching well with the size of the data set. Once the training is over, AS/IJ produces a probability map, which is then thresholded. The corresponding mask is overlaid on the original image as shown in Fig. 2.

For the purposes of comparison we have used the same filters as implemented in Ilastik. For segmentation, we have used two 2D topology-based segmentation metrics i.e. V^{rand} and V^{info} proposed by challenge itself (see next section) [24]. V^{rand} and V^{info} are the normalized version of Rand error and information error after thinning the neurites border [4]. The challenge also found empirically that V^{rand} is more robust than V^{info}. To compute the metrics we have performed 10 fold cross-validation using a random sample of 25 images for training and 5 for testing.

4.3. **Segmentation metrics.** Evaluating results of segmentation is a non-trivial problem, which very much depends on the objective of the analysis. If the objective is correct segmentation of blobs then small errors of the boundary detection will not affect the desired result. In contrast, if pixel-wise error metric is used this situation may result in a large error bond. This has induced some authors to propose non-local or regional metrics. A systematic empirical study of this problem was performed in [3] where the authors recommended specially normalized versions of the "Rand error" V^{rand} and "Variation of Information" V^{info} which best matched expert judgments of segmentation quality for EM structure segmentation.

$$V^{rand} := 2\frac{\sum_{i,j} p_{ij}^2}{\sum_j s_j^2 + \sum_j t_j^2}$$

where $p_{i,j}$ is the probability that a randomly chosen pixel belongs to segment i of the boundary in S (segmented class) and segment j of the boundary in T (ground truth class). $s_i = \sum_j p_{i,j}$ is the marginal distribution density and denotes the probability of that a randomly chosen pixel belongs to segment i in S. $t_j = \sum_i p_{i,j}$ is the co-marginal distribution density and denotes the probability that a randomly chosen pixel belongs to segment j in T.

$$V^{info} := 2\frac{I(S|T)}{H(S) + H(T)}$$

where

$$I(S|T) = \sum_{i,j} p_{ij} \log p_{ij} - \sum_i s_i \log s_i - \sum_j t_j \log t_j$$

Data set	Filters	min scale	max scale
HeLa	LoG, ALoG, Hessian, Structure	2	8
HEp-2	LoG, BoG, ALoG, Hessian, Structure	2	8

TABLE 5. Scale space filters and scales used for Hela and HEp-2 data sets

Regional Feature	Parameters
Legendre	degree (0 – 6)
Zernike	degree (0 –6)
ImageJ	–
Haralick	distance (1-3), directions (90, 180, 270, 360)

TABLE 6. Regional Features and parameters used for Hela and HEp-2 data sets

is the mutual information and

$$H = -\sum p \log p$$

is the corresponding entropy.

4.4. **Image Classification Task.** In order to classify objects, we first enrich the feature space by extracting point features (i.e. LoG, and ALoG) at several scales. The filters, features and parameters used for the task are summarized in Tables 5 and 6. In a third step, we have extracted various regional features (Table 4). Instrumental for the practical applicability of this step is the feature selection, which was performed in the same way as in the segmentation workflow. Next, the Support Vector Machines (SVM) classifier is used to train the model. For the present task, we have used the Sequential Minimal Optimization (SMO) algorithm [32]. The outcome of the classification was evaluated against the ground truth by computing standard metrics – true positive rate (TP), false positive rate (FP), precision, recall, F_1 score, Receiver operating characteristic (ROC) area and Precision-Recall (PR) area.

For classification, we have used 10 fold cross validation and calculated several evaluation metrics in each iteration i.e. true positive (TP), false positive (FP), precision, recall, $F1$ score, area under ROC curve, area under PR curve. The average of all the iterations is reported in Table 8.

5. RESULTS

5.1. **Image Segmentation.** Table 7 shows training and testing results of the AS/IJ and Ilastik. Remarkably, both Ilastik and our platform achieved score 1.0 in V^{rand} and V^{info} metrics on the training data set. On the other hand, Ilastik outperformed AS/IJ by 0.04 in terms of V^{rand} whereas V^{info} was comparable for both platforms.

5.2. **Image Classification.** We have applied the classification based on geometric features on the 3 standard data sets reported here as three case studies.

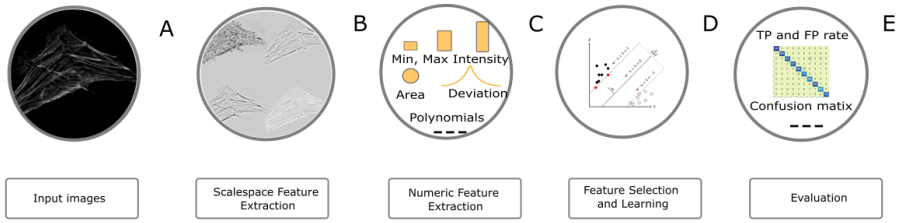

FIGURE 7. Overview of Classification workflow. A: A sample image of filamentous actin labeled with rhodamine-phalloidin (Actin) from HeLa data set; B: Representative images of ALoG, LoG and BoG at scale 4; C: Compound image transforms on original and scaled images, Feature vector is computed; D: CFS based feature selection and SVM's are used training the model; E: Several evaluation metrics are computed e.g. ROC curve, confusion matrix etc.

Metric	Training images		Test images	
	AS/IJ	Ilastik	AS/IJ	Ilastik
V^{rand}	1.0	1.0	0.87	0.91
V^{info}	1.0	1.0	0.93	0.94

TABLE 7. Comparison of AS/IJ and Ilastik on the ISBI data set

FIGURE 8. Confusion matrix of 10 fold cross validation for HeLa and HEp2 using SVM, regional and scale space features and feature selection

5.2.1. *Case study 1: synthetic data set classification.* On the first place, using the synthetic image set AS/IJ was trained with 100 examples of circles and triangles and test it on 50 samples of each. We were able to achieve 100% accuracy R_1 score, area under ROC curve and area under PR curve, using the SVM classifier, for both the test and training data sets. This was to be expected as the data set emphasized the differences in the properties of the Zernike moments compared to the Legendre moments.

5.2.2. *Case study 2: HeLa cells classification.* Classification results were analyzed by computing the confusion matrix. The Fig. 8 demonstrates that there is high confusion between Golgi protein giantin (Golgi) and Golgi protein GPP130 (Golgpp). Note that these protein structures also look similar when inspecting the raw images. The F_1 scores of our method with feature selection is 3 % higher than the results reported by the wnd-charm method [42]. Our approach also achieved 90.2 % area under PR curve and 98.8 % area under ROC curve, respectively. The main advantage of using discriminate features are they optimized well area under PR curve and 2 % more than without feature selection method.

5.2.3. *Case study 3: HEp2 cells classification.* Our classification workflow 4.4 works well even with a large data set. F_1 scores of our method with feature selection is 14 % higher than the results reported by wnd-charm method and very much near

Regional features							
data set	TP	FP	Precision	Recall	F_1-score	ROC area	PR area
Hela	0.82	0.02	0.82	0.81	0.81	0.96	0.75
HEp-2	0.76	0.06	0.77	0.76	0.76	0.90	0.68
Regional features + Scale space pixel-features							
Hela	0.92	0.01	0.92	0.92	0.92	0.98	0.88
HEp-2	–	–	–	–	–	–	–
Regional features + Scale space pixel-features+ discriminative features selection							
Hela	0.93	0.01	0.93	0.93	0.93	0.99	0.90
HEp-2	0.88	0.03	0.88	0.88	0.88	0.95	0.83

TABLE 8. Comparison of mean average true positive rate (TP), false positive rate (FP), precision, recall, F_1 score, ROC area and PR area of different classification approaches in different data sets

to the neural network method[42]. We have also achieved area under 83.0 % PR curve and 95.0 % area under ROC curve respectively.

Table 8 shows the comparison among three different classification scenarios i.e first one uses regional features only and in total we extracted 203 features. The second one is a combination of using regional features and scale space features and third one is also combining feature selection in the second step i.e. 5887 features are extracted for HEp-2 data set. The main problem with incorporating scale space features are the feature space increases exponentially and does not scale well for a large data set like HEp-2 because computation time became excessively long and was skipped. The feature selection step used in the third technique helped to overcome this problem. In total, only 142 and 173 features were selected for HeLa and HEp-2 data sets using the CFS method, respectively. Interestingly, features are selected from both the higher and lower scales. In this regard, we could comment only on the anisotropic Lagrangian ALoG because of its clear geometrical meaning. In case of ALoG, tangential components are selected for some small scales which corresponds to fiber-like structures, while orthogonal components are selected from the larger scale, which corresponds to blob-like structures.

6. DISCUSSION

Presented results can be discussed in several directions.

6.1. **Platform comparison.** The present approach integrates geometrical feature extraction based on signal processing, machine learning and input relying on domain knowledge. The AS/IJ platform is of the same class as two other platforms – TWS and Ilastik. However, there are notable differences in the image processing functionality. Ilastik implements the following filters: ST eigenvalues, Hessian eigenvalues, LoG, Amplitude of Gradient, Difference of Gaussians (DoG) and Gaussian. AS/IJ has more filters implementing scale space theory – notably Bi-Laplacian, Gaussian Jet, Curvatures in 2D and 3D and ALoG. TWS has a more varied filter collection – Gaussian, Hessian, Membrane Projections, ranking filers (mean, max, min, median), variance, entropy, Anisotropic Diffusion, Lipschitz, Gabor, LoG, Sobel, DoG, Kuwahara, Gaussian Derivatives (up to order 5), ST. The TWS platform limits the choice of filters only to hard-coded ones and, therefore, the range of applications

of the platform. Moreover, the platform is geared mostly towards segmentation. AS/IJ is therefore, the platform which implements more completely the functionality prescribed by differential geometry.

A major difference with regard to the other 2 platforms is the interactive feature inspection functionality. In our experience this can be useful in two scenarios. On the frist place, when refining ROIs, one can tailor them to cover filter responses that are more discriminative for the problem. This is facilitated by the functionality to overlay the ROIs on the filter response. On the second place, based on an initial expert choice of filters and the outcome of the initial segmentation, one can decide either to exclude some or include other filters.

6.2. **The problem of ground truth.** The most common bottleneck in most of the segmentation problems is labeling of ground truth for training. In most cases, the ground truth must be provided by experts (i.e. trained neuroanatomists or pathologists). This is a substantial impediment for big-data approaches as ground truth annotation becomes very time-consuming and expensive. Another problem is that for many images it is not possible to delineate with certainty the structures of interest. Therefore, in many cases one has to deal with only partial ground truth. In our view, this is the norm and not the exception. Therefore, any segmentation workflow should explicitly address this issue. Although implicit, such is the understanding when using TWS, which for its time became a landmark innovation in the interactive segmentation[3]. Newer tools, such as Ilastik, also embrace this assumption. In our view, it does not look reasonable that deep learning approaches would immediately alleviate this. One could envision an approach where AS/IJ is used to augment the ground truth provided by experts and use it for training of deep learning CNNs.

6.3. **The problem of feature selection.** Machine learning offers an approach where predefined features are combined into different classifiers. On the other hand, the outcome of machine learning is only as good as the underlying feature space. This can draw some criticism from the proponents of an alternative approach where the features are learned from the image set. The criticism has a point in the sense that simply adding more "hand-crafted" features may not be very useful. On the other hand, one can readily address such criticism by resorting to some generic mathematical theory. The present work supports the view that differential geometry can substantially improve the outcome of machine learning since it can enrich the underlying feature space with new invariant objects. We have used differential geometry to enrich the feature space by adding more integration scales but also have constrained the problem by using only invariant features, which do not depend on the representation. Moreover, these features are linearly independent and, therefore, not trivial to learn from data. The main methodological innovation of the present approach is the use of the orthogonal decomposition of the Laplacian, implemented as the ALoG filter, eqs. 4, 5, which allows for treatment of anisotropic data and to to capture directional information.

In our experience, for blob-like structures, such as nuclei or cytoplasmic vesicles, ALoG alone only gives acceptable segmentation results. For more challenging datasets with varying textures one can use the combinations of ALoG and Hessian or even the full set of filters.

6.4. **Image Classification aspects.** Presented results demonstrate the benefits of using geometric features for image classification in the framework of scale space theory. It is demonstrated that scale and feature selection approaches can overcome the exponential increase of training time that otherwise can be entailed by simply enriching the feature space (Table. 8). Most of the instance classification methods in literature do not have this step, as they directly extract texture features [42]. It should be noted that the SVM classifier is used here merely for comparison purposes. One can also use any other classifier available in the Weka library.

6.5. **Outlook.** The present paper discusses only four limited use cases where comparison with other platforms helps situate the presented work. However, this by no means exhausts the possible use of AS/IJ. An interesting future direction of research using the platform will be explore feature spaces and compare them to intermediate outcomes of CNNs where image-driven features are computed. This could potentially offer more insights into the operation of CNNs. This can be achieved by comparisons with platforms, such as Cellpose [43] and StarDist [39].

Another interesting direction of research, which is currently in progress, is to investigate α-scale spaces and to also make available differential invariants based on α-scale space theory[14].

Acknowledgments

The development of Active Segmentation platform was supported in part by grants from Research Fund - Flanders (FWO), (contract numbers, G.0C75.13N, VS.097.16N), The COST action INDEPTH (CA 16212), as well as Google Summer of Code 2016 – 2021, where the projects were sponsored by INCF and the Belgian Neuroinformatics Node in the scope of the ImageJ working group `https://incf.org/sig/imagej-working-group`. In particular, we would like to acknowledge the efforts of Victor Jose Garcia Fernandez, Belgian Neuroinformatics Node intern 2015; Mukesh Gupta, GSOC 2017; Sanjeev Dubej, GSOC 2018; Raghavendra Singh Chauhan, GSOC 2020; Joanna Stachera, GSOC 2020.

References

[1] M.D. Abramoff, P.J. Magalhaes, and S.J. Ram. Image Processing with ImageJ. *Biophotonics International*, 11(7):36–42, 2004.

[2] L. Alvarez, F. Guichard, P. L. Lions, and J. M. Morel. Axioms and fundamental equations of image processing. *Archive for rational mechanics and analysis*, 123(3):199–257, 1993.

[3] I. Arganda-Carreras, V. Kaynig, C. Rueden, K. W. Eliceiri, J. Schindelin, A. Cardona, and S. H. Seung. Trainable Weka Segmentation: a machine learning tool for microscopy pixel classification. *Bioinformatics (Oxford, England)*, March 2017.

[4] I. Arganda-Carreras, S. C. Turaga, D. R. Berger, D. Cireşan, A. Giusti, L. M. Gambardella, J. Schmidhuber, D. Laptev, S. Dwivedi, J. M. Buhmann, T. Liu, M. Seyedhosseini, T. Tasdizen, L. Kamentsky, R. Burget, V. Uher, X. Tan, C. Sun, T. D. Pham, E. Bas, M. G. Uzunbas, A. Cardona, J. Schindelin, and H. S. Seung. Crowdsourcing the creation of image segmentation algorithms for connectomics. *Frontiers in Neuroanatomy*, 9, nov 2015.

[5] P. Bankhead, M. B. Loughrey, J. A. Fernández, Y. Dombrowski, D. G. McArt, P. D. Dunne, S. McQuaid, R. T. Gray, L. J. Murray, H. G. Coleman, J. A. James, M. Salto-Tellez, and P. W. Hamilton. QuPath: Open source software for digital pathology image analysis. *Scientific Reports*, 7(1), dec 2017.

[6] S. Basu, S. Kolouri, and G. K. Rohde. Detecting and visualizing cell phenotype differences from microscopy images using transport-based morphometry. *PNAS*, 111(9):3448–3453, 2014.

[7] S. Berg, D. Kutra, T. Kroeger, C. N. Straehle, B. X. Kausler, C. Haubold, M. Schiegg, J. Ales, T. Beier, M. Rudy, K. Eren, J. I Cervantes, B. Xu, F. Beuttenmueller, A. Wolny, C. Zhang, U. Koethe, F. A. Hamprecht, and A. Kreshuk. ilastik: interactive machine learning for (bio)image analysis. *Nature Methods*, 16(12):1226–1232, sep 2019.

[8] M. V. Boland and R. F. Murphy. A neural network classifier capable of recognizing the patterns of all major subcellular structures in fluorescence microscope images of HeLa cells . *Bioinformatics*, 17(12):1213–1223, 12 2001.

[9] T. Brox, J. Weickert, B. Burgeth, and P. Mrazek. Nonlinear structure tensors. Technical report, Universitat des Saarlandes, 2004.

[10] A. Cardona, S. Saalfeld, S. Preibisch, B. Schmid, A. Cheng, J. Pulokas, P. Tomancak, and V. Hartenstein. An integrated micro- and macroarchitectural analysis of the drosophila brain by computer-assisted serial section electron microscopy. *PLoS Biology*, 8(10):e1000502, October 2010.

[11] C.-C. Chang and C.-J. Lin. LibSVM. *ACM Transactions on Intelligent Systems and Technology*, 2(3):1–27, April 2011.

[12] C. Cortes and V. N. Vapnik. Support vector networks. *Machine Learning*, 20:273 – 297, 1995.

[13] F. de Chaumont, S. Dallongeville, N. Chenouard, N. Hervé, S. Pop, T. Provoost, V. Meas-Yedid, P. Pankajakshan, T. Lecomte, Y. Le Montagner, T. Lagache, A. Dufour, and J-C Olivo-Marin. Icy: an open bioimage informatics platform for extended reproducible research. *Nature Methods*, 9(7):690–696, jun 2012.

[14] R. Duits, M. Felsberg, L. Florack, and B. Platel. Alpha-scale spaces on a bounded domain. scale space methods in computer vision. *494- 510. Springer*, 2003.

[15] R. E. Fan, K-W Chang, C-J Hsieh, X.-R. Wang, and C-J Lin. Liblinear: A library for large linear classification. In *JMLR*, 2008.

[16] L. Florack, B. Ter Haar Romeny, M. Viergever, and J. Koenderink. The Gaussian scale-space paradigm and the multiscale local jet. *International Journal of Computer Vision*, 18(1):61–75, apr 1996.

[17] L. M J Florack, B. M ter Haar Romeny, J. J Koenderink, and M. A Viergever. Scale and the differential structure of images. *Image and Vision Computing*, 10:376–388, 1992.

[18] P. Geurts, A. Irrthum, and L. Wehenkel. Supervised learning with decision tree-based methods in computational and systems biology. *Molecular BioSystems*, 5(12):1593, 2009.

[19] M. A. Hall. Correlation-based feature selection for discreate and numeric class machine learning. *University of Waikato, Hamilton, NewZealand*, 1999.

[20] R. M. Haralick, K. Shanmugam, and I. Dinstein. Textural features for image classification. *IEEE Transactions on Systems, Man, and Cybernetics*, SMC-3(6):610–621, nov 1973.

[21] T. K. Ho. Random decision forests. In *Proceedings of the 3rd International Conference on Document Analysis and Recognition, Montreal, QC, 14–16 August 1995*, pages 278–282, 1995.

[22] P. Hobson, B. C. Lovell, G. Percannella, M. Vento, and A. Wiliem. Classifying anti-nuclear antibodies hep-2 images: A benchmarking platform. In *2014 22nd International Conference on Pattern Recognition*, pages 3233–3238, 2014.

[23] T. Iijima. Theory of pattern recognition. *Electronics and Communications in Japan*, pages 123 – 134, 1963.

[24] V. Jain, B. Bollmann, M. Richardson, D. R. Berger, M. N. Helmstaedter, K. L. Briggman, W. Denk, J. B. Bowden, J. M. Mendenhall, W. C. Abraham, K. M. Harris, N. Kasthuri, K. J. Hayworth, R. Schalek, J. C. Tapia, J. W. Lichtman, and S. Seung. Boundary learning by optimization with topological constraints. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. IEEE, June 2010.

[25] J. J. Koenderink. The structure of images. *Biological Cybernetics*, 50(5):363–370, aug 1984.

[26] Michael R. Lamprecht, David M. Sabatini, and Anne E. Carpenter. CellProfiler^{TM}: free, versatile software for automated biological image analysis. *BioTechniques*, 42(1):71–75, jan 2007.

[27] A. B. L. Larsen, S. Darkner, A. L. Dahl, and K. S. Pedersen. Jet-based local image descriptors. In *Computer Vision – ECCV 2012*, pages 638–650. Springer Berlin Heidelberg, 2012.

[28] T. Lindeberg. Scale-space. *Wiley Encyclopedia of Computer Science and Engineering*, pages 2495–2504, 2007.

[29] D. Marr and E. Hildreth. Theory of edge detection. *Proc. R. Soc. Lond. B, 207:187-217*, 207:187 – 217, 1980.

[30] E. Meijering. Cell segmentation: 50 years down the road [life sciences]. *IEEE Signal Processing Magazine*, 29(5):140–145, sep 2012.

[31] RF Murphy, MV Boland, and M. Velliste. Towards a systematics for protein subcelluar location: quantitative description of protein localization patterns and automated analysis of fluorescence microscope images. *Proc Int Conf Intell Syst Mol Biol*, 251-9:187 – 217, 2000.

[32] J. C. Platt. Sequential minimal optimization: A fast algorithm for training support vector machines. Technical report, ADVANCES IN KERNEL METHODS - SUPPORT VECTOR LEARNING, 1998.

[33] A. Poulet, I. Arganda-Carreras, D. Legland, A. V. Probst, P. Andrey, and C. Tatout. NucleusJ: an ImageJ plugin for quantifying 3d images of interphase nuclei. *Bioinformatics*, 31(7):1144–1146, nov 2014.

[34] D. Prodanov, T. Konopczynski, and M. Trojnar. Selected applications of scale spaces in microscopic image analysis. *Cybernetics and Information Technologies*, 15(7):5–12, dec 2015.

[35] D. Prodanov and K. Verstreken. *Molecular Imaging*, chapter Automated Segmentation and Morphometry of Cell and Tissue Structures. Selected Algorithms in ImageJ, pages 183 –208. InTech, 2012.

[36] X. Qi, G. Zhao, J. Chen, and M. Pietikäinen. Exploring illumination robust descriptors for human epithelial type 2 cell classification. *Pattern Recognition*, 60:420–429, December 2016.

[37] C. T. Rueden, J. Schindelin, M. C. Hiner, B. E. DeZonia, A. E. Walter, E. T. Arena, and K. W. Eliceiri. ImageJ2: ImageJ for the next generation of scientific image data. *BMC Bioinformatics*, 18(1), nov 2017.

[38] J. Schindelin, I. Arganda-Carreras, E. Frise, V. Kaynig, M. Longair, T. Pietzsch, S. Preibisch, C. Rueden, S. Saalfeld, B. Schmid, J-Y Tinevez, D. James White, V. Hartenstein, K. Eliceiri, P. Tomancak, and A. Cardona. Fiji: an open-source platform for biological-image analysis. *Nature Methods*, 9(7):676–682, jun 2012.

[39] U. Schmidt, M. Weigert, C. Broaddus, and G. Myers. Cell detection with star-convex polygons. In *Medical Image Computing and Computer Assisted Intervention MICCAI 2018*, pages 265–273. Springer International Publishing, 2018.

[40] C. A Schneider, W. S Rasband, and K. W Eliceiri. NIH Image to ImageJ: 25 years of image analysis. *Nature methods*, 9:671–675, July 2012.

[41] S. Sezgin and B. Sankur. Survey over image thresholding techniques and quantitative performance evaluation. *J. Electron. Imaging 13(1): 146-165*, 2004.

[42] M. Shifat-E-Rabbi, X. Yin, C.E. Fitzgerald, and G.K. Rohde. Cell image classification: A comparative overview. *Cytometry Part A*, 97(4):347–362, feb 2020.

[43] Carsen Stringer, Tim Wang, Michalis Michaelos, and Marius Pachitariu. Cellpose: a generalist algorithm for cellular segmentation. *Nature Methods*, 18(1):100–106, December 2020.

[44] T. Vicar, J. Balvan, J. Jaros, F. Jug, R. Kolar, M. Masarik, and J. Gumulec. Cell segmentation methods for label-free contrast microscopy: review and comprehensive comparison. *BMC Bioinformatics*, 20(1), jun 2019.

[45] J. Weickert. Anisotropic diffusion in image processing. *ECMI Series. Teubner-Verlag, Stuttgart,*, 1998.

[46] A.P. Witkin. Scale-space fltering. *In Proc. 8th Int. Joint Conf. Artificial Intelligence (IJCAI 83), volume 2, 1019-1022*, 1983.