RESEARCH ARTICLE

# A Benchmark Data Set to Evaluate the Illumination Robustness of Image Processing Algorithms for Object Segmentation and Classification

Arif ul Maula Khan, Ralf Mikut, Markus Reischl*

Institute for Applied Computer Science, Image and Data Analysis Group, Karlsruhe Institute of Technology, Karlsruhe, Baden-Wuerttemberg, Germany

* markus.reischl@kit.edu

## Abstract

Developers of image processing routines rely on benchmark data sets to give qualitative comparisons of new image analysis algorithms and pipelines. Such data sets need to include artifacts in order to occlude and distort the required information to be extracted from an image. Robustness, the quality of an algorithm related to the amount of distortion is often important. However, using available benchmark data sets an evaluation of illumination robustness is difficult or even not possible due to missing ground truth data about object margins and classes and missing information about the distortion. We present a new framework for robustness evaluation. The key aspect is an image benchmark containing 9 object classes and the required ground truth for segmentation and classification. Varying levels of shading and background noise are integrated to distort the data set. To quantify the illumination robustness, we provide measures for image quality, segmentation and classification success and robustness. We set a high value on giving users easy access to the new benchmark, therefore, all routines are provided within a software package, but can as well easily be replaced to emphasize other aspects.

## Introduction

Image processing is a means for automatically extracting image contents, often used in science (e.g. biological readouts (mouse [1], fish [2], insect [3]), surveying and mapping [4], particle accelerator [5] and man-machine interaction [6, 7]) or industry (nutrition industry [8, 9], quality supervision [10] or pick-and-place applications [11, 12], an overview of image processing in industry is given in [13].).

In computer vision applications, image processing routines need to be developed for image data sets containing a set of similar images. This happens in many real-time acquisition systems (surveillance camera etc.), lab equipment (high-throughput microscopic imaging etc.) or

offline analysis of big databases containing similar images (objects in satellite images, human blood cells analysis in laboratory etc.).

In general, an image processing routine consists of the elements preprocessing and filtering, segmentation, interpretation and quantification, each consisting of further sub-units or operators building a so-called pipeline. The segmentation seeks to assign each pixel a property (e.g. being part of a structure or not), the interpretation assigns pixels with same properties to similar objects and the quantification assigns each object a feature vector of numbers describing the properties of each object. In some cases, classification algorithms are applied to assign a label to all found objects based on the object feature vector.

Each step of the pipeline contains parameters which are tuned by the developer. The outcome of the segmentation is highly dependent on the correct use of both, sub-routines and parameters. It depends, therefore, on the experience of the developer. Correctly parametrized pipelines deliver good results on the image data set they were designed for, however they often fail on unknown data.

To develop and evaluate new algorithms in comparison with standard methods, routines and parameter sets are validated using benchmark data sets with a clear ground truth about objects. Therefore, numerous benchmark data sets exist (e.g. data sets used in [14] for benchmarking and validation in biological image segmentation, [15] for cell image analysis by using simulated cell populations, [16] for semantic automatic image annotations using complex scenes, [17] for event recognition in surveillance videos and many others). Images in certain data sets contain highly complex objects or background as in large data sets e.g. [18–20] such as animals and vehicles against a complex background. Benchmark databases for traffic signs [21], a street scene with pedestrians, a lion sitting on grass, visual event recognition in videos etc. also exist. In other cases, there are sets with incomplete ground truth http://www.broadinstitute.org/bbbc/ such as images containing cells with only the total number of cells as ground truth without any information of the type and shape of the objects to be segmented.

Major problems in image processing are alterations in image data sets due to effects like shading, noise etc. Since different images deliver different results, algorithms are hard to compare. Therefore, not only the absolute quality of a segmentation algorithm for validation data sets is important, but also its ability to cope with unknown distortions, called robustness. To evaluate robustness, the segmentation quality in relation to the intensity of the alteration is a good measure. To derive this measure for a new routine, the benchmark must not only contain the ground truth for segments but also image variations of aforementioned effects. Furthermore, the strength of these effects needs to be given for each sample. Benchmarks known in literature do not deliver this data. Therefore, it is not possible to evaluate new image processing algorithms for robustness.

Furthermore, if the subsequent outcome of a classifier is to be evaluated not only a ground truth in segmentation but also in object labels is required. For classifier validation, there are plenty of benchmark data sets (IRIS [22] or WINE [23]). Benchmarks allowing image segmentation and object classification in combination with a quantified distortion have not yet been published. As well, there are robustness and evaluation measures for image segmentation [24], but no consistent methodology to evaluate both, the combination of image segmentation and classification.

This article introduces a new benchmark data set for the validation of robustness in image processing and classification. It contains easy-to-find segments of different object types (having different shapes, sizes, mean intensity values etc.) with given ground truth in segmentation and classification. Distortions like shading, noise and overlapping objects are inserted in varying intensities. Altogether, we introduce four data sets, each containing labeled segments of 5–9

object classes. The data sets contain a series of images with increasing shading intensity and noise level.

The distortion is quantified by calculating a fuzzy artifact level depending on amount of shading and intensity of noise. A measure for algorithm quality and outcome is provided by three measures, quantifying segmentation quality, segmentation overlap and classification accuracy.

To evaluate new image processing routines efficiently, we introduce an exemplary image processing pipeline and show how to apply the new framework. We prove the functionality comparing two basic image processing routines and deliver initial results on the data set regarding accuracy and robustness. The sources for the image processing routine, the robustness measures and the data set as well are freely available and downloadable https://sourceforge.net/projects/gait-cad/files/Benchmarks/hardware_items/.

## Materials and Methods

### Benchmark data set—irregular shaped solid hardware items

Details about data sets and information about object types are given in Table A in S1 Text. In order to evaluate the robustness, the image benchmark data set itself requires a quantification of image quality called artifact level and quality measures for segmentation and object classification.

The benchmark consists of images $\mathbf{X} = ((x_{ij})) \in \mathbb{N}^{m \times n}$ with $m = 1000$, $n = 1500$, $x_{ij} \in [0, 255]$ with variations in brightness/shading and noise. There are $R = 4$ data sets, each combining $B = 13$ grades of shading and $N = 14$ grades of noise. Thus, altogether $B \cdot R \cdot N = 728$ images are contained in the data set (see Fig 1). Images are saved as 8bit TIFF-files, the naming convention is 'benchdata_r_b_n.tif', where $r$, $b$, $n$ are data set number, brightness and noise level respectively. Each number is represented as two digits (e.g. 'benchdata_02_01_03.tif').
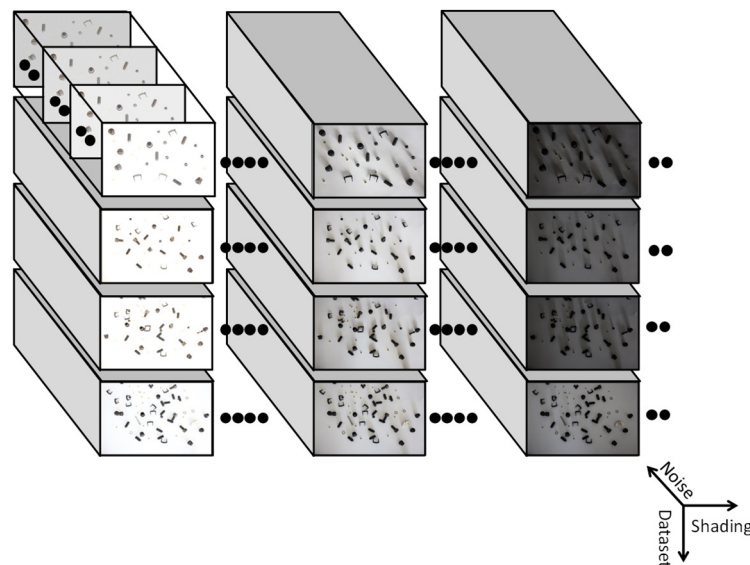


**Fig 1. Benchmark data sets.** Overview of images in $R = 4$ scenes corresponding to increasing artifact and noise levels

doi:10.1371/journal.pone.0131098.g001
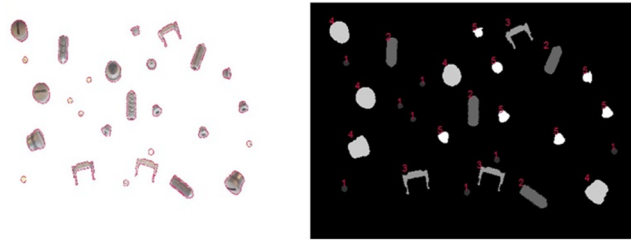
**Fig 2. Representation of ground truth objects using color shading and numbers as labels.** Left: Brightfield image **X** of a benchmark data set with marked object edges, right: Ground truth image **X**$_{truth}$, with object types in numbers. Gray scales denote the value of $x_{ij,truth}$ (0: black (background), 5: white)

Shadowing and shading were introduced by two light sources from above and aside. Images were taken using a professional camera with a remote trigger in order to keep object positions same for $b = 1, \ldots, B$ lighting conditions ($b = 1$: very bright, no shading; $b = 13$: dark, lot of shading).

For each acquired image, artificial background noise in varying intensities was added. Altogether, 13 additional noise levels were generated using Gaussian-distributed random numbers with varying standard deviations. Thus, the image $\mathbf{X}(r, b, n)$ denoting data set $r$, brightness level $b$ and noise level $n$ is given by:

$$x_{ij}(r, b, n) = f(x_{ij}(r, b, n) + x_{\mathrm{rnd},ij}), \tag{1}$$

where $x_{\mathrm{rnd},ij}$ is a realization of the Gaussian-distributed random variable $X_{\mathrm{rnd},ij} \sim N(0, \sigma_n^2)$ with $\sigma_n^2 = 125(n-1)$. The function $f$ restricts possible values to $[0, 255]$:

$$f(x) = \begin{cases} 0 & \text{if} & x < 0 \\ 255 & \text{if} & x > 255. \\ x & \text{else} \end{cases} \tag{2}$$

The maximum number of objects in any data set is denoted as $O$ and total class types are denoted as $K$. Objects are not aligned and multiple viewing directions are possible. For each image, a ground truth in object classification and segmentation is given: $\mathbf{X}_{\text{truth}} = ((x_{ij}))_{\text{truth}} \in \mathbb{N}^{m \times n}$ with $x_{ij,\text{truth}} \in \{0, \ldots, K\}$ (0: background, 1, ..., $K$: objects). An exemplary image is depicted in Fig 2.

The ground truth is provided using full lighting conditions and semi-automatic labeling (first using Otsu segmentation [25] and then manual correction in Windows Paint Application), both in terms of their type and object boundaries (see Fig 2). See S2 Text for more details on manual labeling.

## Measures for image quality, segmentation and robustness

To evaluate the quality of a new image processing routine, on the one hand, image quality needs to be quantified, on the other hand segmentation and classification success needs to be measured. In this section, we briefly describe the motivation of all measures. The use of the given measures provides direct applicability for other users but is not mandatory and can also be replaced by tailor-made or more general pipelines. Therefore, all routines and evaluation measures are given in source code that can be downloaded and even manipulated.

To enable algorithm developers to incorporate robustness into their algorithms, we provided benchmark images with distortions and artifacts in varying strengths. The amount of distortion is measured by an artifact function $A(r, b, n)$ which aggregates the shading level and the noise level. Shading is quantified using the mean pixel value in an image, noise is quantified by the parameter $\sigma_n$ of the Gaussian-distributed noise. A monotonic fuzzification function $\mu \in [0, 1]$ based on the quantifying parameter delivers a mean to normalize the effects of shading and noise. $\mu$ contains tunable higher and lower bounds $\alpha$ and $\beta$, to suppress upper and lower outliers as shown in Fig A of S3 Text. Detailed descriptions are given in S3 Text.

To evaluate the outcome of an image processing routine, we use a quality criterion $Q(r, b, n)$ based on the number of detected objects (Segmentation measure 1), their respective areas (Segmentation measure 2) and the classification accuracy based on the number of misclassified objects (Classification measure). It is specifically designed to incorporate further evaluation measures in addition to the necessary criteria for segmentation evaluation. Segmentation measure 1 compares the detected objects to the objects given in the ground truth and thus quantifies the outcome of object recognition. Segmentation measure 2 evaluates the pixel positions of the detected objects related to the positions of ground truth objects. It judges the quality of the foreground/background discrimination. The Classification measure is based on the difference between assigned classes by the classifier in comparison to true object classes. The total quality measure $Q(r, b, n)$ fuzzifies and combines the three evaluation measures. The S4 Text contains all procedures in detail.

Using the aforementioned measures we derive a quality measure for robustness $R$. If the quality measure $Q(r, b, n)$ is plotted over the artifact level $A(r, b, n)$, algorithm 1 is called to be more robust, if its data-points exceed the data-points of algorithm 2 in their values. This is usually quantified using the area under the curve or by adapting a tansig-regression and evaluating the position of the inflection point.

To have an objective set of varying artifact levels for performance evaluation, we select images constituting of an image series $\mathbf{X}(r, k, k + 1)$ where $k = 1, \ldots, 13$. An algorithm is then assigned the robustness measure

$$R = \frac{\sum_{b=1}^{B} Q(r, b, b + 1)}{B}.$$ (3)

## Exemplary processing pipeline

We implemented an exemplary pipeline for standard algorithms given in Fig 3 to apply the aforementioned methods and thus to compare two or more subroutines in an image processing pipeline. This pipeline was used for testing standard algorithms by evaluation their performance on the benchmark data sets.

In this pipeline, a grayscale image is first normalized using 2% and 98% quantiles. The normalized image is then segmented using standard image segmentation techniques. Here, we want to use the benchmark to compare a standard Otsu operator to a Sobel edge detector [26]. The resulting binary image is then passed through morphological operations such as image hole filling, image opening and image border clearing in order to get reasonable objects. Each object is described by 8 features to design a Bayes classifier. Features are: area, mean intensity, maximum intensity, minimum intensity, solidity, median intensity, standard deviation of intensity values and eccentricity. Eccentricity is defined as the ratio of the distance between the ellipse foci and its major axis length. It is a scalar value between 0 and 1 where 0 is a circle and 1 is a line segment. This pipeline has been implemented as macro in the Gait-CAD software ([27]) developed in MATLAB and is provided with the implementation code.

## 1 Input grayscale image
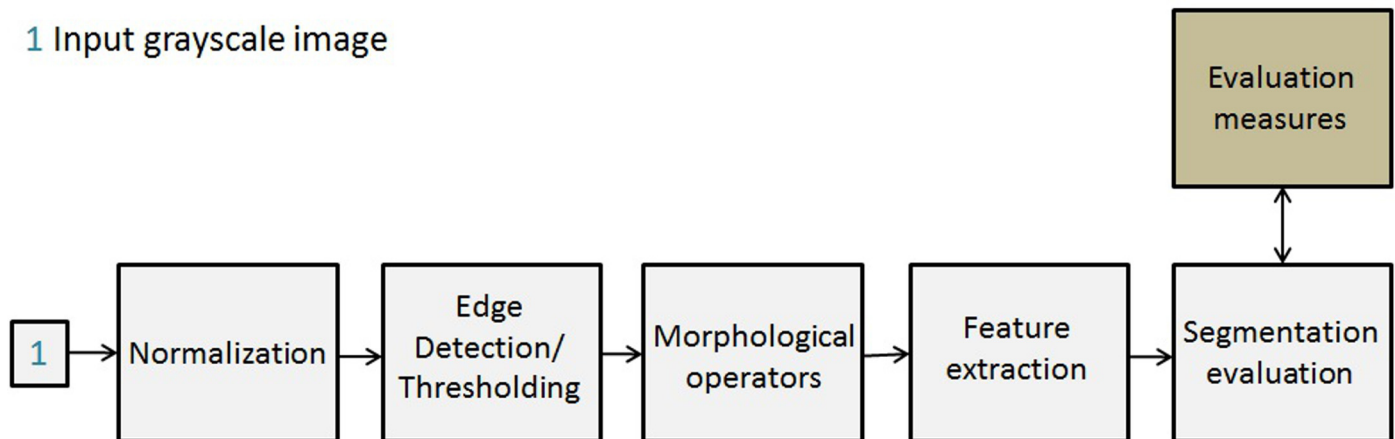


**Fig 3. Exemplary pipeline for the segmentation of benchmark images.**

doi:10.1371/journal.pone.0131098.g003

## Results

We use the image processing pipeline given in Fig 3 to compare the segmentation steps i.e. Otsu operator versus a Sobel edge detector. As a native edge detection does not directly seek for objects, we expect the edge detector to deliver worse results. However, the subsequent hole filling delivers objects in undistorted cases. The robustness of each pipeline is quantified using Eq (3) and Data Set $r = 1$. This scene $r = 1$ was selected due to the lower number of foreground objects and less complex cases to make this demonstration more understandable.

The performance of the two pipelines, one using the Otsu operator and the other using the Sobel edge detector with respect to increasing artifact levels are given in Fig 4. The segmentation evaluation measures are integrated in total quality measure $Q(r, b, n)$.

Fig 4 shows a decreasing performance of the segmentation algorithms indicated by $Q(r, b, n)$ with increasing artifact levels ($A(r, b, n)$). To quantify the performance degradation, the robustness measures $R_{otsu}$ and $R_{edge}$ for Otsu thresholding and Sobel edge detection, respectively are given. $R$ is given on a scale between 0 and 1 where larger values show high robustness. Both segmentation methods are found to be comparatively closer to each other. At higher artifact levels, edge detection fails to find the adequate number of objects as compared to Otsu thresholding but classification results are seen to be better than those obtained with Otsu segmentation. Wrong classification assignments occur if the artifact level is increased.

## Discussion

The presented framework contains images of the same scenes under varying illumination conditions and noise levels as well as the ground truth for segment detection and object type classification. Furthermore, we provide measures to evaluate artifact level, segmentation and classification quality. Thus, robustness evaluation of image processing and classification algorithms becomes possible and enables developers to compare image processing algorithms with respect to robustness. To this end, the development of algorithms with a focus on robustness in distorted data is accelerated. Parameters and structures can easily be evaluated and optimized.
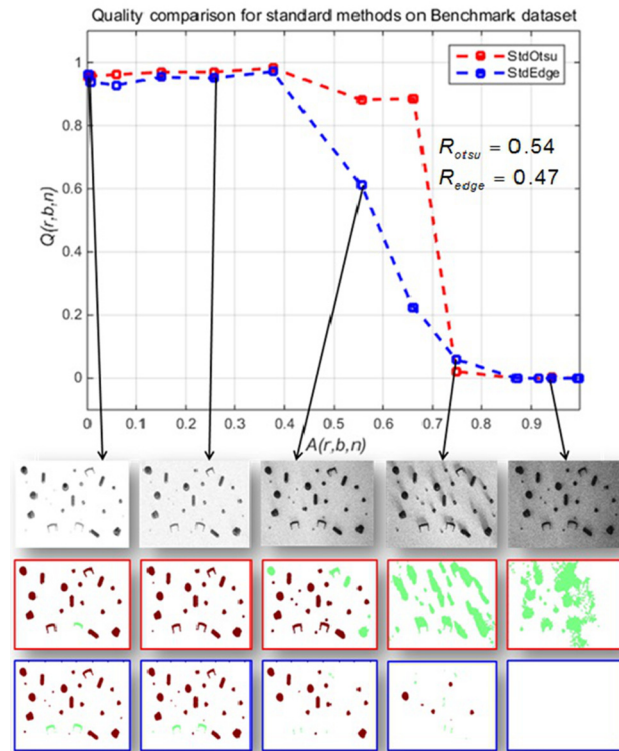
**Fig 4. Results of benchmark data set r = 1.** Total quality criterion $Q(r, b, n)$ vs. increasing artifact level $A(r, b, n)$ for image series with stepwise addition of both shading and noise for each successive image. The first row of images indicates original images from data set $r = 1$. The second row shows corresponding segmentation and classification results using Otsu's method. The third row shows corresponding segmentation and classification results using edge detection method. Brown color represents correct classification of the segmented BLOB w.r.t corresponding ground truth BLOB and light green color shows an erroneous classification object. Robustness values for Otsu thresholding and Sobel edge detection are $R_{otsu}$ = 0.54 and $R_{edge}$ = 0.47 respectively.

doi:10.1371/journal.pone.0131098.g004

Furthermore, not only the robustness of an image-processing pipeline can be evaluated but also segmentation quality. Parameters can be optimized with respect to given data quality. For example, if the quality of images is known to be bad, algorithms do not need to provide optimal results on good quality images.

In addition, the outcome of an image processing pipeline can be fed back to optimize its parameters. Therefore, users have a means to tune parameters in a pipeline not only to fit a special set of images but to be applied to a more general class of problems.

Basically each part of the framework (images, artifact level calculation, quality calculation, robustness) can individually be replaced, depending on the preferences of the developer. Even the images may be replaced by arbitrary images as long as the calculation of the artifact level (depending on the parameter to be used, here: mean pixel value) delivers reasonable results.

We provided all measures and routines in source code to allow other developers a uniform processing to compare their algorithms to standard algorithms.

Using the proposed benchmark, we have shown the effect of increasing artifact levels on the image segmentation outcome using standard algorithms. The proposed quality measures may be used for other object classification benchmark without a special robustness focus as well.

In our ongoing work, we use the presented framework to parametrize and evaluate new feedback-based image processing routines.

## Supporting Information

**S1 Text. Data and object information. Table A**. Distribution of total object number $O$ and maximum class type $O$ in different scenes.
(PDF)

**S2 Text. Data Labeling.**
(PDF)

**S3 Text. Artifact level. Fig A**. Spline based fuzzy artifact function.
(PDF)

**S4 Text. Quality measures for segmentation and classification.**
(PDF)

## Acknowledgments

## Author Contributions

Conceived and designed the experiments: AK MR RM. Wrote the paper: AK MR RM. Developed evaluation criteria: AK MR. Applied pipelines to benchmark: AK.

## References

1. Zhang W, Morris Q, Chang R, Shai O, Bakowski M, Mitsakakis N, et al. (2004) The functional landscape of mouse gene expression. Journal of Biology 3: 21. doi: 10.1186/jbiol16 PMID: 15588312

2. Lieschke GJ, Currie PD (2007) Animal models of human disease: zebrafish swim into view. Nature Reviews Genetics 8: 353–67. doi: 10.1038/nrg2091 PMID: 17440532

3. Nüsslein-Volhard C, Wieschaus E (1980) Mutations affecting segment number and polarity in drosophila. Nature 287: 795–801. doi: 10.1038/287795a0 PMID: 6776413

4. Caselles V, Kimmel R, Sapiro G (1997) Geodesic active contours. International Journal of Computer Vision 22: 61–79. doi: 10.1023/A:1007979827043

5. Bock R, Grote H, Notz D, Regler M (2009) Data Analysis Techniques for High-Energy Physics Experiments. Academic Press.

6. Boehme HJ, Baumann U, Corradini A, Gross H (1999) Person localization and posture recognition for human-robot-interaction. In: Braffort A, editor, Gesture Based Communication in Human-Computer Interaction, Springer Verlag.

7. Nickel K, Stiefelhagen R (2007) Visual recognition of pointing gestures for human–robot interaction. Image and Vision Computing 25: 1875–1884. doi: 10.1016/j.imavis.2005.12.020

8. Brosnan T, Sun D (2004) Improving quality inspection of food products by computer vision—a review. Journal of Food Engineering 61: 3–16. doi: 10.1016/S0260-8774(03)00183-3

9. Du C, Sun D (2004) Recent developments in the applications of image processing techniques for food quality evaluation. Trends in Food Science and Technology 15: 230–249. doi: 10.1016/j.tifs.2003.10.006

10. Kavoussanos M, Pouliezzos A (1998) An intelligent robotic system for depalletizing and emptying polyethylene sacks. International Journal of Advanced Manufacturing Technology 14: 358–362. doi: 10.1007/BF01178915

11.  Patil G, Chaudhari D (2013) Sift based approach: Object recognition and localization for pick-and-place system. International Journal of Advanced Research in Computer Science and Software Engineering 3 (3): 196–201.

12.  Wang WH, Liu XY, Sun Y (2006) Robust contact detection in micromanipulation using computer vision microscopy. In: Proc., 28th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBS). pp. 2219–2222.

13.  Malamas E, Petrakis E, Zervakis M, Petit L, Legat J (2003) A survey on industrial vision systems, applications and tools. Image and Vision Computing 21(2): 171–188. doi: 10.1016/S0262-8856(02)00152-X

14.  Gelasca ED, Byun J, Obara B, Manjunath B (2008) Evaluation and benchmark for biological image segmentation. In: IEEE International Conference on Image Processing (ICIP). pp. 1816–1819.

15.  Ruusuvuori P, Lehmussola A, Selinummi J, Rajala T, Huttunen H, Yli-Harja O (2008) Benchmark set of synthetic images for validating cell image analysis algorithms. In: Proceedings of the 16th European Signal Processing Conference, EUSIPCO.

16.  Escalante HJ, Hernández CA, Gonzalez JA, López-López A, Montes M, Morales EF, et al. (2010) The segmented and annotated IAPR TC-12 benchmark. Computer Vision and Image Understanding 114: 419–428. doi: 10.1016/j.cviu.2009.03.008

17.  Oh S, Hoogs A, Perera A, Cuntoor N, Chen CC (2011) A large-scale benchmark dataset for event recognition in surveillance video. In: Proc., IEEE Conference on Computer Vision and Pattern Recognition (CVPR). IEEE, pp. 3153–3160.

18.  Guthaus MR, Ringenberg JS, Ernst D, Austin TM, Mudge T, Brown RB. (2001) MiBench: A free, commercially representative embedded benchmark suite. In: Proc., IEEE International Workshop on Workload Characterization (WWC-4). IEEE, pp. 3–14.

19.  Deng J, Dong W, Socher R, Li L, Li K, Fei-Fei L (2009) Imagenet: A large-scale hierarchical image database. In: Proc., IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Ieee, pp. 248–255.

20.  Ge F, Wang S, Liu T (2007) New benchmark for image segmentation evaluation. Journal of Electronic Imaging 16: 033011-1–033011-16.

21.  Stallkamp J, Schlipsing M, Salmen J, Igel C (2011) The German traffic sign recognition benchmark: A multi-class classification competition. In: Proc., International Joint Conference on Neural Networks (IJCNN). IEEE, pp. 1453–1460.

22.  Anderson E (1935) The irises of the Gaspé Peninsula. Bulletin of the American Iris Society 59: 2–5.

23.  Forina, M et al (1991). PARVUS—an extendible package for data exploration, classification, and correlation. Institute of Pharmaceutical and Food Analysis and Technologies, Via Brigata Salerno, 16147 Genoa, Italy.

24.  Stegmaier J, Otte JC, Kobitski A, Bartschat A, Garcia A, Mikut R, et al. (2014) Fast segmentation of stained nuclei in terabyte-scale, time resolved 3D microscopy image stacks. PLoS ONE 9: e90036. doi: 10.1371/journal.pone.0090036 PMID: 24587204

25.  Otsu N (1979) A threshold selection method from gray-level histograms. IEEE Transactions on Systems, Man and Cybernetics 9: 62–66. doi: 10.1109/TSMC.1979.4310076

26.  Sobel I, Feldman G (1968) A $3 \times 3$ isotropic gradient operator for image processing.

27.  Stegmaier J, Alshut R, Reischl M, Mikut R (2012) Information fusion of image analysis, video object tracking, and data mining of biological images using the open source MATLAB toolbox Gait-CAD. Biomedizinische Technik (Biomedical Engineering) 57 (S1): 458–461.