Automatic Object Detection using DBSCAN for Counting Intoxicated Flies in the FLORIDA Assay

Christian Bodenstein*‡, Markus Götz*‡, Annika Jansen†, Henrike Scholz†, Morris Riedel*‡
* Juelich Supercomputing Center, Leo-Brandt-Straße, 52425 Juelich, Germany
{c.bodenstein, m.goetz, m.riedel}@fz-juelich.de
† University of Cologne, Department of Biology, Zuelpicher Straße 47b, 50674 Cologne, Germany
{annika.jansen, henrike.scholz}@uni-koeln.de
‡ University of Iceland, Sæmundargötu 2, 101 Reykjavik, Iceland

***Abstract*—In this paper, we propose an instrumentation and computer vision pipeline that allows automatic object detection on images taken from multiple experimental set ups. We demonstrate the approach by autonomously counting intoxicated flies in the FLORIDA assay. The assay measures the effect of ethanol exposure onto the ability of a vinegar fly *Drosophila melanogaster* to right itself. The analysis consists of a three-step approach. First, obtaining an image of a large set of individual experiments, second, identify areas containing a single experiment, and third, discover the searched objects within the experiment. For the analysis we facilitate well-known computer vision and machine learning algorithms—namely color segmentation, threshold imaging and DBSCAN. The automation of the experiment enables an unprecedented reproducibility and consistency, while significantly decreasing the manual labor.**

***Index Terms*—Image Analysis, Computer Vision, Machine Learning, FLORIDA Assay, DBSCAN, Biology, Flies, Genetics**

I. Introduction

Object detection is one of the fundamental problems in computer vision. It is concerned with identifying objects in an image or video irrespective of variations to it, like for example different viewpoints, scaling, rotation, translation or partial obstruction. The literature proposes a multitude of approaches to tackle the problem, such as feature matching, template matching or machine learning. In this paper we propose a method that is based on the latter, namely, the unsupervised clustering algorithm DBSCAN and threshold imaging.

We demonstrate the approach by counting intoxicated vinegar flies *Drosophila melanogaster*. The data is generated with the FLORIDA assay, which measures the effect on ethanol onto the ability to right itself again after intoxication. This is used to identify genes and neuronal mechanisms underlying the intoxication effect. While we focus on the automation of the FLORIDA assay, the pipeline could be generalized to similar detection problems e.g. bacterial culture observation on well plates.

The remainder of this paper is structured as follows. Section II reviews related work, and Section III presents background information on the FLORIDA assay and the used algorithms. In sections IV and V the analysis process is presented in detail along with its implementation. An evaluation of the approach based on experimentation data is discussed in section VI and, finally, the paper is concluded in section VII followed bv an outlook on improvements in section VIII.

II. Related Work

Clustering is an established method for detecting and segmenting objects in images and videos. Some of the earlier attempts date back to the 1970's and 1980's. Coleman et al. [7] for instance propose an image segmentation method based on K-Means [13], one of the fundamental clustering algorithms, that is able to split a complete image into segments in the respective color space. Similar research has been conducted by Haralick et al. [14] in 1981 which is based upon hierarchical clustering and its different linkage types [13]. A good summary of these earlier attempts is given by Kettaf et. al [16].

In biology and bio-medicine K-Means, and its adaptive and fuzzy variants, are in widespread use, because of its robustness to low image resolution and noise in the images. Examples include tumor detection [18], volumetric reconstruction of the left ventricle chamber [6] or automatic identification of brain regions on MRT images [21]. More recently, density-based approaches have become popular. Unlike K-Means based approaches, they are able to directly detect non-circular objects in images, without having to piece them together from a set of sub-clusters, while at the same not requiring an exact number *k* of clusters to be identified. Celebi et al. [5] for instance apply DBSCAN [10] to segment an arbitrary amount of irregular skin lesions in dermatological imagery. For our problem at hand both of the above properties are desirable as we, first, do not know the number of individual objects, second, observe objects of arbitrary shapes. A more generalized method for density-based object segmentation in images is given by Ye et al [24] upon which we have based our approach.

III. Background

This section provides an overview of the experimental set up for data generation and used algorithms.

A. FLORIDA Assay

The "**F**ull **L**oss **O**f **R**ighting Reflex **In**Duced by **A**lcohol" - FLORIDA assay is used to identify genes and neuronal networks underlying the ethanol induced loss of righting reflex, a measure for the degree of intoxication. The experimental animal is the common vinegar fly *Drosophila melanogaster*, that is a useful genetic tool to study the mechanistic basis of behaviors associated with alcoholism [22]. The vinegar flies not only share common genes but also behavioral similarities when intoxicated, and are therefore a useful genetic model system to understand ethanol induced behaviors in humans. Through alteration of genes in the specimen and subsequent observation of mutants, it is possible to analyze changes of behaviors that are associated with alcohol abuse disorders [22]. Concretely, the experiment is focused on the investigation of the flies' tolerance to the intoxication effect of ethanol. In line with that, a population of flies is exposed to vaporizing ethanol in experimentation vials. The individuals' intoxication level is recorded over time, by counting sedated flies. The sedation is defined as flies that fail to right themselves after a mechanic stimulation. This measure is also known as loss of righting reflex. The experimentation procedure is summarized as follows:

1) Multiple individuals, currently a group of 20, are placed into an experimentation vial. 2) The vial is closed by an ethanol soaked tissue which vaporizes over time. 3) In a specified interval—currently 60 seconds—the vials are shaken to startle the flies and identify the sedated flies that fail to right themselves. 4) The number of intoxicated objects needs to be determined in an image. 5) The experiments is terminated when all flies are sedated. 6) The data analysis has to be obtained for multiple vials containing groups of flies that are treated in parallel.

B. DBSCAN

Clustering algorithms in the field of machine learning are used to aggregate similar objects into common groups. *DBSCAN* is a particular, density-based clustering algorithm that was published 1996 by Ester et al. [10]. Its principal idea is to find dense areas and to expand these recursively in order to find clusters. A dense region is thereby formed by a point that has within a given search radius ϵ at least θ neighboring points, whereas θ is called density threshold, or in literature often referred as $minPoints$.

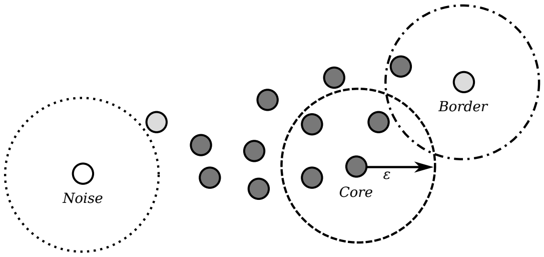


Fig. 1. $DBSCAN$ clustering with $minPoints$ $\theta = 4$ and search radius ε.

This dense area is also called the *core* of a cluster. For each of the found neighbor points the density criteria is reapplied and the cluster is consequently expanded. All points that do not form a cluster core and that are not "absorbed" through expansion are regarded as *noise*. A more formal definition can be found in [10]. Two of the major advantages of DBSCAN compared to other traditional clustering algorithms, like K-Means [13] for example, is that it can detect arbitrarily shaped clusters without having to know the number of clusters apriori.

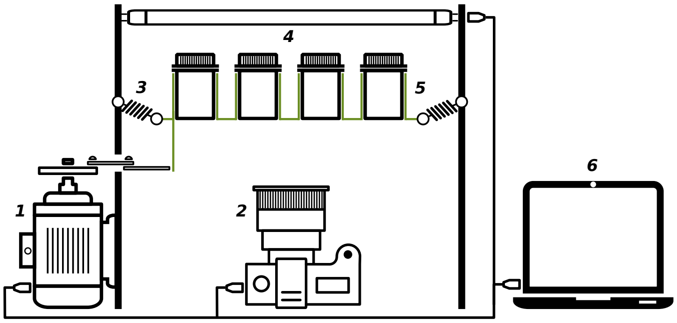


Fig. 2. The automated experimentation setup. It is consisting of (1) an electric engine, shaking the vials via a rotary crank, (2) a digital SLR, (3) a spring suspension allowing horizontal motion, (4) a top lightning plate, (5) the experimentation vials containing the fruit flies and (6) a notebook with the analysis software, connected via USB to other devices.

IV. Analysis Process

Counting the number of intoxicated flies is achieved in a three-step sequential analysis process. First, a high resolution image is retrieved from the FLORIDA laboratory setup's single-lens reflex (SLR) camera. Then, the vials are segmented to enable individual fly counting for each vial. Finally, after thresholding the image, the DBSCAN algorithm is used to cluster the darker image areas. The result is then used to count the number of flies in the image. Details of these steps are described in the sections below.

A. The FLORIDA laboratory setup

The FLORIDA laboratory setup, illustrated in Figure 2, has been built in order to enable the fully automatic counting of sedated flies and consists of several sub-parts. The main part is the holder plate for the experimentation vials, attached to the outer framework through metal springs which allow horizontal motion. The holder plate exhibits a grid of four by five vial slots, permitting a analysis of 20 vials in parallel with each vial equally holding 20 *drosophila melanogaster*.

The bottom of the holder plate is plastered with green foil, enabling accurate segmentation of the vial slots. A SLR camera is placed below the holder plate, with the lens faced to the bottom of the vials. Sedates flies gathering at the vials' bottoms can easily be captured by the camera, while flies in upper regions fade out due to the opaque vial material. To achieve a high contrast between flies and vial surface, a LED light plate is placed above the holder plate to screen the vials. As shown in Figure 2, an electronic engine is connected to the holder plate, to shake the vials before the photo is taken as to test the righting reflex. The SLR camera, the shaker engine and the light plate are connected via USB to a PC and can be controlled with our FLORIDA software. This includes capturing and streaming images from the camera directly to the program.

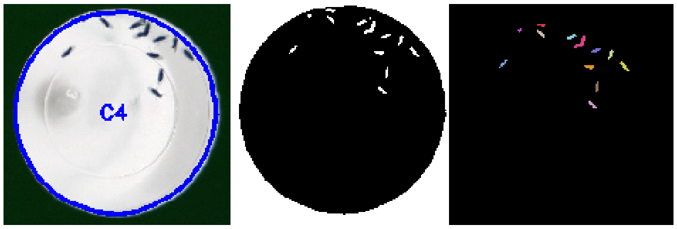


Fig. 3. The image processing steps from left to right—left: vial segmentation; center: threshold image; right: resulting clusters.

B. Vial Segmentation

Before the actual flies can be detected, the vials have to be found. Our first attempt was to use the Hough Transform [8] for finding circles. While this method was able to find the vials quickly, it does not provide the desired precision. Perspective distortions of the camera let the vials shapes appear more elliptical than circular. This is a problem that is difficult, while not impossible to manage with this algorithm.

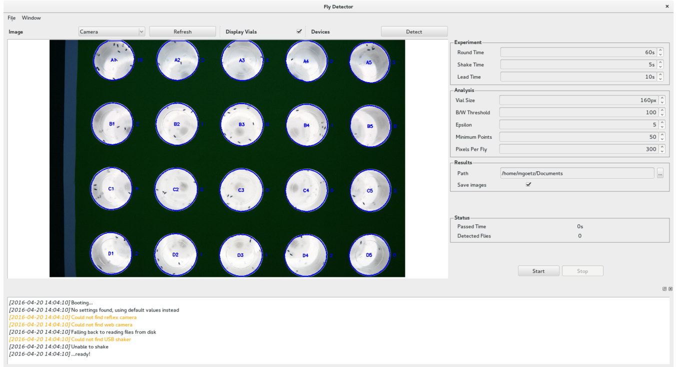
To provide fast and accurate vial segmentation a green foil was glued on the underneath of the holder plate. This enables the usage of simple color segmentation techniques based on green screening (similar to what is done in movies). For this, the image has to be converted from RGB (Red Green Blue) to HSV (Hue Saturation Value) [11] color space, since it separates the color information from intensity and illumination. The hue value describes the color which should be in range $h = [40, 80]$ for green, the saturation, or color intensity, should be at least in range $s = [150, 255]$ and the value, or illumination is between $v = [10, 255]$, to avoid the segmentation of complete dark regions. After a mask is created that selects all pixels within these ranges, contours are detected using the Teh-Chin chain approximation algorithm [23]. All contours that contain an area of around $a = \pi\psi^2$, where ψ is the predefined vial radius, are considered as vials.

The resulting vials mask is post-processed with an erosion algorithm to smooth the edges. Furthermore, the contour points are moved few pixels towards the image center where outer points are stronger affected than inner points. This *rubber band effect* has the purpose to correct the different perspective view angles on the vials. The results are stored as polygons which can be used to create an image mask for the associated vial.

C. Threshold Clustering

After the vial segmentation step, each vial can now be analyzed separately. For this, the RGB image is converted first into a gray scale image and then the vial mask is used to whiten everything but the considered vial. Then, a threshold τ is defined. Given the image width X and image height Y, all pixels $p_{x,y}$ in the resulting image P are binarized using the following function:

$$p_{x,y} = \begin{cases} 255 & \text{if } p_{x,y} \leq \tau \\ 0 & \text{if } p_{x,y} > \tau \end{cases} \forall x \in [1, X] \forall y \in [1, Y] \qquad (1)$$



Experiment

Round Time: 60s

Shake Time: 5s

Lead Time: 10s

Analysis

Vial Size: 160px

B/W Threshold: 100

Epsilon: 5

Minimum Points: 50

Pixels Per Fly: 300

Results

Path: /home/mgoetz/Documents

Save images ✓

Status

Passed Time: 0s

Detected Flies: 0

```
[2016-04-20 14:04:10] Booting...
[2016-04-20 14:04:10] No settings found, using default values instead
[2016-04-20 14:04:10] Could not find reflex camera
[2016-04-20 14:04:10] Could not find web camera
[2016-04-20 14:04:10] Falling back to reading files from disk
[2016-04-20 14:04:10] Could not find USB shaker
[2016-04-20 14:04:10] Unable to shake
[2016-04-20 14:04:10] ...ready!
```

Fig. 4. The graphical user interface of the FLORIDA software with the to be analyzed image on the left and parameter settings on the right

This means, all pixels below the threshold are considered to belong to a fly and are mapped to white, while all other pixels are mapped to black. The coordinates of the white pixels W are then extracted:

$$W = \{(x, y) | \forall x \in [1, X] \forall y \in [1, Y] : p_{x,y} > 0\} \qquad (2)$$

These are then fed to DBSCAN:

$$C, N = \text{DBSCAN}(\epsilon, \theta, W) \qquad (3)$$

Whereby C are the resulting cluster labels and N the pixels considered to be noise.

D. Fly Counting

Finally, the found clusters C have to be counted. The naïve approach is to consider each cluster to represent a single fly. However, due to being close to one another, a single cluster may contain multiple flies, as seen in Figure 3, this will result in miscounts. Therefore, the fly count $\hat{f}$ is predicted as such:

$$\hat{f} = \sum_{i}^{|C|} \left\lceil \frac{|C_i|}{\rho} \right\rceil \qquad (4)$$

With ρ being an additional pixels per fly parameter, defining the amount of cluster pixels ($|C_i|$) that will count as single fly. While more complex algorithms have been considered to overcome the problem, this solution is fast and yields sufficient performance. The threshold clustering and fly counting steps are repeated for each found vial in the image.

V. Implementation

The entire analysis process is implemented as part of a standalone GUI application that is operated by the laboratory assistants of the FLORIDA experiment. Figure 4 depicts an example of the interface, while the next section describe details.

A. Controls

The interface of the FLORIDA software is divided into five major parts. On top of the window is a menu bar that allows to load and store experimentation settings, if the defaults saved on the last exit are undesired. Below that, is another menu bar that is focused on controlling the external devices. It allows to automatically detect and set up the external devices and take a fresh picture with the camera. Moreover, the currently displayed picture analysis mode—that is raw, threshold or clustering—can be selected here as well as vial detection rendering toggled. The actual image is displayed in the center of the GUI window.

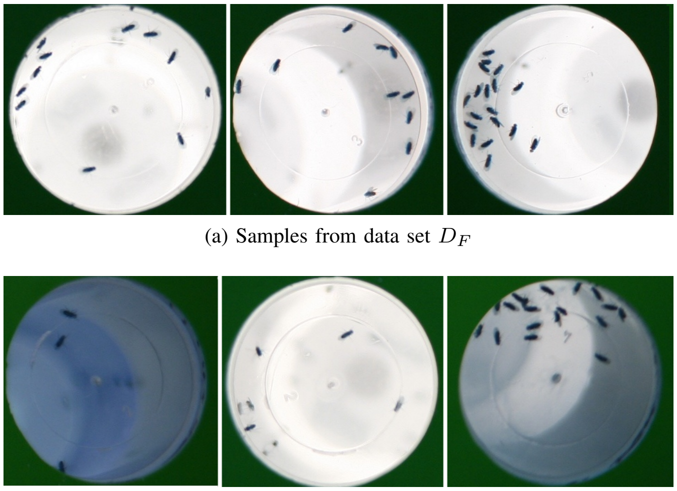
On the right side of the window is a pane with the experimentation settings. Here, the FLORIDA laboratory assistants can configure analysis parameters, like ϵ, θ, ρ, device properties, like the shake time and its lead, as well as result options, like e.g. the path, where the results are stored in a comma-separated value format. On the bottom of the main application window a system log is displayed. It informs the user about the software's status, which includes among other things which devices are connected and which not. The logger can be closed at anytime in order to enlarge the image and can be re-opened via the top menu.

B. Software

The source code of the software can be obtained from the authors' public Github repository [3] and is licensed under BSD-style restrictions, meaning it can be used and modified free of charge. It is written in C++ with the help of the Qt programming framework [1] and the interface has been designed using its built-in creator suite.

In order to communicate with the external devices and to analyze the images the FLORIDA software requires other software dependencies, all of them being open-source as well. For obtaining images from the SLR libgphoto2 [15] is utilized. This library enables camera control automation including image streaming to computers. Supporting a large set of models from common vendors allows a transparent exchange of the actual SLR. The shaker, or electric engine, is remote controlled via an USB relay that is programmed using libusb [9], implementing only the required packages to turn it on or off. Due to the use of a standardized communication protocol for the relay, an exchange of the USB relay is possible.

Internally, the FLORIDA software employs two software packages to process the images. On the one hand, we have OpenCV [4] providing convenience functions for image editing, like color space conversion, cropping, threshold and so forth. On the other hand, a stripped down OpenMP-version of HPDBSCAN [12]—a parallel processing variant of the regular DBSCAN—is used, to fully utilize the computer's processing capabilities. An installation script for the FLORIDA software can be found in the Github repository, effectively obtaining the newest version, compiling it and installing it, including setting correct access rights for the devices.



(a) Samples from data set D_F

(b) Samples from data set D_N

Fig. 5. Samples from two different FLORIDA experiments. While D_F was contains high quality images, the data from D_N differs in terms of illumination and resolution.

C. Hardware

In the current FLORIDA experimentation setup a dual-core laptop with 2.0 GHz and 2 GB of RAM is controlling the other devices. On it is the Fedora 21 operating system installed. The pictures are taken by a Canon EOS 5D Mark I SLR camera, which has a native resolution of 4368×2912 pixels and full frame image sensor format. The high resolution provides enough image quality to record 20 vials concurrently. The vials are shaken by a electric engine that is switched on and off by a "single channel 5V USB Control switch".

VI. Evaluation

An important aspect of each machine learning pipeline is to measure its performance. For the FLORIDA experiment, dedicated validation data sets have been collected and the model parameter were compared to each other by a well defined loss function. The results of these steps can be found in this section.

A. Datasets

To make an evaluation possible, two datasets have been recorded. The first data set D_N is obtained from an accelerated FLORIDA experiment. Instead of capturing one image each 60 seconds, the interval has been decreased to 15 seconds to increase the number of resulting images. At the point when all flies were sedated and after the removal of poor quality images a total number of 1381 vials image were gathered. This dataset still includes low quality pictures in terms of illumination and noise that should not occur in an actual analysis environment. However, we decided to keep this images in the dataset to give machine learning algorithms the chance to be more sensitive to these outliers. The second data set D_F was recorded in the regular one minute interval over 44 minutes total, resulting in a collection of 880 vials. While D_N will be considered as a benchmark dataset to build more robust models, D_F is

TABLE I
GRID SEARCH RANGES—GRAND TOTAL OF 39,900 COMBINATIONS

	Min	Max	Step
Threshold (τ)	90	120	5
Epsilon (ϵ)	2	11	1
MinPts (θ)	1	19	1
PixelsPerFly (ρ)	100	390	10

considered as real experimental data set. All recorded vials have been counted manually and are available on the research repository B2SHARE [2].

B. Parameter search

As a next step, the parameters for the algorithms needed to be found. While the vial size ψ can be easily determined by measuring it on the images, the parameters τ (threshold), ϵ (DBSCAN search radius), θ (DBSCAN density threshold) and ρ (pixels per fly) are more difficult to estimate and are highly dependent on each other.

First, we introduce a user experience parameter set (UEPS) which is simply a set of parameters that have been defined by laboratory assistants, while experimenting with the FLORIDA software and that provide sufficient enough counting accuracy. These parameters are:

$$\text{UEPS} = \begin{bmatrix} \tau \\ \epsilon \\ \theta \\ \rho \end{bmatrix} = \begin{bmatrix} 100 \\ 5 \\ 15 \\ 200 \end{bmatrix} \tag{5}$$

While the UEPS represents just the manual users subjective perception, an automated grid search about a certain range of parameter sets has been performed. With help of the JURECA [17] supercomputer, the grid search over $39,900$ total parameter sets could be computed in parallel. In Table I one can find the parameter ranges that have been searched.

The grid search sorts the parameter sets ascending by the mean squared error (MSE) value on D_F. We also provide the R^2 value, which is another efficient metric to compare the sum of squared errors to the variance of the data.

$$R^2 = 1 - \frac{SSE_f}{V_f} = 1 - \frac{\sum_i (f_i - \hat{f}_i)^2}{\sum_i (f_i - \bar{f})^2} \tag{6}$$

Where f_i is the actual number of flies $\hat{f}_i$ the predicted number of flies and $\bar{f}$ the mean number of flies. This value tends to one, if the model perfectly predicts the desired outcome and is zero, if the prediction is equals to the mean. Additionally, the MSE and R^2 values were computed for D_N. The top five parameter sets with respect to their MSE on D_F can be found in Table II.

C. Interpretation

The gird search shows that small values for ϵ and θ are favorable and that a threshold τ around 90 yields the best results. Moreover, ρ tends to be smaller than initially guessed in the UEPS. Given that the values of ϵ and θ are set below to a certain range the error increases drastically, as one can see in Figure 6.

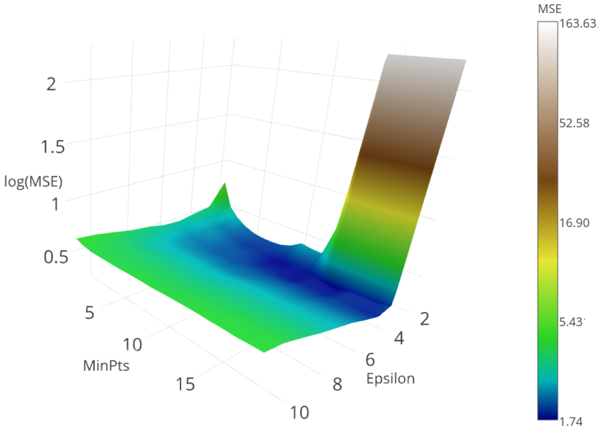


Fig. 6. Three-dimensional projection of the grid search. The DBSCAN parameters ϵ and θ are compared to the resulting MSE. The dark blue regions contain the lowest MSE values.

As expected, the MSE and R^2 values are significantly higher for the D_N data set. A possible explanation is the different illumination in the data, where extreme dark regions could be perceived as large fly clusters, or flies with low contrast would not be recognized as such. In order to handle this problem, one would require a dynamical threshold to binarize the image, depending on its light properties. Interestingly enough, the top five parameter sets with the highest pixels per fly value ρ result in the lowest MSE on D_N. This underlines the statement from above, that large ρ values will result in a lower MSE for accidentally misclassified fly clusters.

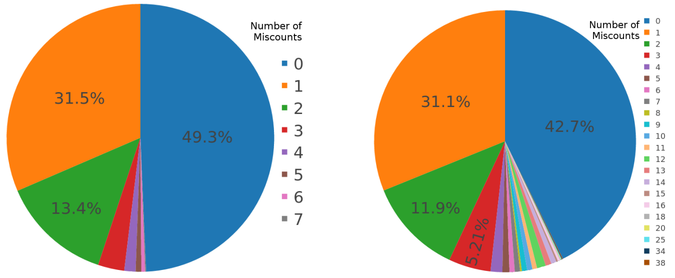
The MSE of 1.745 on D_F is equal to a standard deviation of 1.32 miscounted flies on all images. The results in Table II and the flat grid search surface in Figure 6 indicate that the most optimal parameter sets are similar to one another and robust enough for experimentation conditions. Therefore, we suggest to use the best found parameter set, further referred as grid search parameter set (GSPS) which is:

$$\text{GSPS} = \begin{bmatrix} \tau \\ \epsilon \\ \theta \\ \rho \end{bmatrix} = \begin{bmatrix} 90 \\ 3 \\ 9 \\ 160 \end{bmatrix} \tag{7}$$

Furthermore, in case of deviating experiment environments, new parameters can be found by using the GSPS as a basis.

TABLE II
GRIDSEARCH TOP FIVE RESULTS

#	τ	ϵ	θ	ρ	MSE$_{D_F}$	$R^2_{D_F}$	MSE$_{D_N}$	$R^2_{D_N}$
1	90	3	9	160	1.745	0.946	12.433	0.727
2	90	3	9	150	1.750	0.946	12.912	0.717
3	95	3	15	160	1.757	0.946	19.233	0.578
4	90	3	9	200	1.768	0.946	11.355	0.751
5	90	3	10	160	1.770	0.946	12.559	0.724
UEPS	100	5	15	200	3.325	0.899	26.732	0.414



(a) Error Distribution D_F

(b) Error Distribution D_N

Fig. 7. Absolute miscounting errors made by our model, using the grid search parameter set (GSPS).

The MSE performance increase of the GSPS over the initial UEPS is 190% for D_F and 215% for D_N. While this is an significant improvement, the MSE may not provide enough information to validate how accurate the model counts flies. By listing the occurrences of miscounted flies in Figure 7, the reader should get an impression of the model's performance. In the D_F data set, nearly half of the vials has been counted correctly, while another 45% of the vials were miscounted by one to two flies. Only five percent show up higher error rates with up to seven miscounts. The performance in D_N shows that the number of correctly counted flies is lower, but the main drop in performance comes from a low number of extremely high errors. As mentioned before, this could be attributed to dark regions that are misclassified as flies.

In summary the performance of the model exceeds our expectations. Nevertheless, for dark and noisy images a more robust model is required. For well prepared, equally illuminated experiment environments the accuracy of the model is sufficient.

VII. Conclusion

In this paper we presented an automatic image analysis pipeline that allows the counting of intoxicated fruit flies in the FLORIDA assay. We have proposed both, a experimentation setup for devices as well as the analyzing algorithms. The latter was implemented in a GUI application that is currently in productive use at the University of Cologne. In an empirical evaluation of the software we were able to obtain an MSE of 1.745. An adaptation of the method to similar segmentation and counting problems in lab environments is possible through the adjustment of a few model parameters.

VIII. Future Work

More robust models lead to generalization for more scientific domains. One idea is to use the generated fly masks from our model, to train a segmentation convolutional neural network as proposed in [19]. With the help of data augmentation, i.e. modifying illumination and rotation, a robust model could be trained, without the need for an explicit threshold parameter tuning. Another approach is to train a convolution neural network through reinforcement learning similar to what Mnih et al. [20] described, in order to dynamically adapt the models parameters to the image. Finally, image classification could be used on the found clusters, to discover multiple flies in one cluster. This step would need the creation of an additional data set, providing images of found clusters and labeled with the number of visible flies. This paper will be followed by biological experiments at the University of Cologne.

References

[1] J. Blanchette and M. Summerfield. *C++ GUI programming with Qt 4*. Prentice Hall Professional, 2006.

[2] C. Bodenstein. Florida sample data. http://hdl.handle.net/11304/29fb9a58-1558-40c0-b3e3-2642b12f8e70. [Data Set].

[3] Bodenstein, Christian and Götz, Markus. FlyDetector Github repository. https://github.com/cbodenst/FlyDetector. [Online, accessed 21-04-2016].

[4] G. Bradski and A. Kaehler. *Learning OpenCV: Computer vision with the OpenCV library*. " O'Reilly Media, Inc.", 2008.

[5] M. E. Celebi, Y. A. Aslandogan, and P. R. Bergstresser. Mining biomedical images with density-based clustering. In *Information Technology: Coding and Computing, 2005. ITCC 2005. International Conference on*, volume 1, pages 163–168. IEEE, 2005.

[6] C. W. Chen, J. Luo, and K. J. Parker. Image segmentation via adaptive k-mean clustering and knowledge-based morphological operations with biomedical applications. *Image Processing, IEEE Transactions on*, 7(12):1673–1683, 1998.

[7] G. B. Coleman and H. C. Andrews. Image segmentation by clustering. *Proceedings of the IEEE*, 67(5):773–785, 1979.

[8] Duda and Hart. Use of the hough transformation to detect lines and curves in pictures. *Communications of the ACM*, 15(1):11–15, 1972.

[9] Erdfelt, J and Drake, D. libusb. http://www.libusb.org/. [Online, accessed 21-04-2016.

[10] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Kdd*, volume 96, pages 226–231, 1996.

[11] A. Ford and A. Roberts. Colour space conversions. *Westminster University, London*, 1998:1–31, 1998.

[12] M. Götz, C. Bodenstein, and M. Riedel. Hpdbscan: highly parallel dbscan. In *Proceedings of the Workshop on Machine Learning in High-Performance Computing Environments*, page 2. ACM, 2015.

[13] J. Han, M. Kamber, and J. Pei. *Data Mining: concepts and techniques - 3rd ed.* Morgan Kaufmann, 2011.

[14] R. M. Haralick and L. G. Shapiro. Image segmentation techniques. *Computer vision, graphics, and image processing*, 29(1):100–132, 1985.

[15] Hubert Figuière. libgphoto2. http://www.gphoto.org/proj/libgphoto2/. [Online, accessed 21-04-2016.

[16] Kettaf, Bi, and Beauville. A comparison study of image segmentation by clustering techniques. In *Signal Processing, 1996., 3rd International Conference on*, volume 2, pages 1280–1283. IEEE, 1996.

[17] D. Krause and P. Thörnig. JURECA: General-purpose supercomputer at Jülich Supercomputing Centre. *Journal of large-scale research facilities*, 2:A62, 2016.

[18] B. N. Li, C. K. Chui, S. Chang, and S. H. Ong. Integrating spatial fuzzy clustering with level set methods for automated medical image segmentation. *Computers in biology and medicine*, 41(1):1–10, 2011.

[19] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3431–3440, 2015.

[20] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, and M. Riedmiller. Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*, 2013.

[21] H. Ng, S. Ong, K. Foong, P. Goh, and W. Nowinski. Medical image segmentation using k-means clustering and improved watershed algorithm. In *Image Analysis and Interpretation, 2006 IEEE Southwest Symposium on*, pages 61–65. IEEE, 2006.

[22] Scholz and Mustard. Invertebrate models of alcoholism. In *Behavioral neurobiology of alcohol addiction*, pages 433–457. Springer, 2011.

[23] C.-H. Teh and R. T. Chin. On the detection of dominant points on digital curves. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 11(8):859–872, 1989.

[24] Q. Ye, W. Gao, and W. Zeng. Color image segmentation using density-based clustering. In *Acoustics, Speech, and Signal Processing, 2003. Proceedings.(ICASSP'03). 2003 IEEE International Conference on*, volume 3, pages III–345. IEEE, 2003.