IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2017.DOI

Grid Screener: A Tool for Automated High-throughput Screening on Biochemical and Biological Analysis Platforms

MARCEL P. SCHILLING[1], SVENJA SCHMELZER[1], JOAQUÍN EDUARDO URRUTIA GÓMEZ[2], ANNA A. POPOVA[2], PAVEL A. LEVKIN[2], and MARKUS REISCHL[1]

[1]Karlsruhe Institute of Technology, Institute for Automation and Applied Informatics, Hermann-von-Helmholtz-Platz 1, 76344 Eggenstein-Leopoldshafen (e-mail: {marcel.schilling,svenja.schmelzer,markus.reischl}@kit.edu)
[2]Karlsruhe Institute of Technology, Institute of Biological and Chemical Systems, Hermann-von-Helmholtz-Platz 1, 76344 Eggenstein-Leopoldshafen (e-mail: {anna.popova,joaquin.gomez,pavel.levkin}@kit.edu)

Corresponding author: Marcel P. Schilling (e-mail: marcel.schilling@kit.edu).

This work was funded in the KIT Future Fields project "Screening Platform for Personalized Oncology (SPPO)" and was partly performed on the HoreKa supercomputer funded by the Ministry of Science, Research, and the Arts Baden-Württemberg and by the Federal Ministry of Education and Research. Besides, this work is supported by the Helmholtz Association Initiative and Networking Fund on the HAICORE@KIT partition. This project was partly supported by DFG (Heisenbergprofessur Projektnummer: 406232485, LE 2936/9-1). Furthermore, we thank the Helmholtz Program "Materials Systems Engineering" for the support.

ABSTRACT Grid structures are common in high-throughput assays to parallelize experiments in biochemical or biological experiments. Manual analysis of grid images is laborious, time-consuming, expensive, and critical in terms of reproducibility. However, it is still common to do such analysis manually, as there is no standardized software for automated analysis. In this paper, we introduce a generic method to automatically detect grid structures in images and to perform flexible spot-wise analysis after successful grid detection. The deep learning-based approach of the grid structure detection allows being flexible concerning different grid types. The combination with a robust parameter estimation algorithm lowers the requirements of the detection quality and thus enhances robustness. Further, the method conducts semi-automated grid detection if a fully automated processing fails. An open-source software tool Grid Screener that implements the proposed methods is provided as a ready-for-use tool for researchers. The usability is demonstrated by taking different criteria into account, which are important for a successful application. We present the benefits of our proposed tool Grid Screener utilizing three different grid types in the context of high-throughput screening to show our contribution towards further lab automation. Our tool performs much faster than manual analysis, while maintaining or even enhancing accuracy.

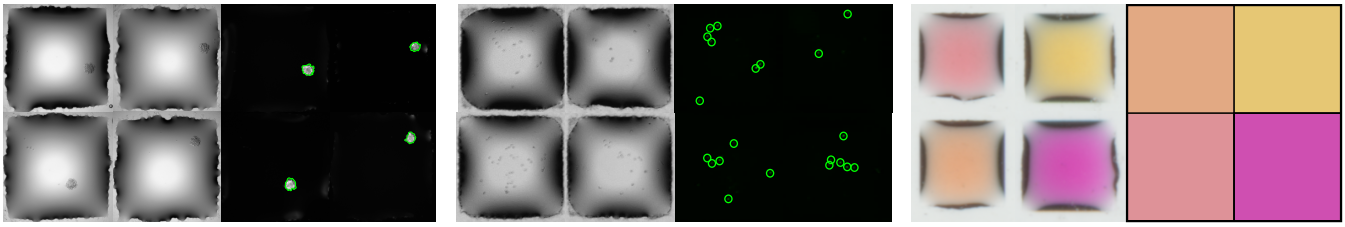
INDEX TERMS Application software, Artificial neural networks, Automation, Biological systems, Chemical technology, Machine learning, Parameter estimation, Image processing

I. INTRODUCTION

High-throughput assays have become an indispensable tool for modern biotechnology and biology [1]–[3] since they enable generating, analysis, and processing of a large amount of data. Using assays, overall experimental run times can be significantly reduced while improving data reliability and reproducibility by eliminating human error. One of the major tools in today's biology and chemistry to enhance high-throughput is miniaturization and parallelization of assays. By reducing working volumes up to a million-fold, a superior level of spatiotemporal control [4] is provided which allows to work with a wider range of samples and experimental conditions whilst reducing operational costs [2].

Assays on miniaturized platforms such as drug screening [5], bacterial drug colorimetric assays [6], analysis of blood vessels [7], and embryoid body screening [8] rely primarily on visualization of the phenomenon by microscopy. Due to the design of these platforms, the images often have grid-like structures composed of hundreds to thousands of spots, from which the information relevant to the experiment

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3135709, IEEE Access



(a) Image Segmentation (b) Cell Detection (c) Colorimetric Analysis

FIGURE 1: **Overview of different spot-wise processing:** Exemplary tasks can be the segmentation (a, right) and cell detection (b, right) in fluorescent images or the colorimetric analysis (c, left) of digital images. Segment contours (a, right) and regions of interest (b, right) are marked in green. Further, median RGB color values are extracted and given (c, right). In addition, bright-field microscopy images (left) are presented in the case of (a) and (b).

needs to be analyzed.

Often, the experiment's outcome is given as fluorescent and bright-field microscope images [9] or color digital images [10]. Examples are presented in Figure 1. For instance, the segmentation of spheroids on fluorescent images (cf. Figure 1a), counting cells per spot in fluorescent images (cf. Figure 1b), or analysis of color digital images (cf. Figure 1c) are possible tasks within the high-throughput image screening.

Thereby, grid structures of different spots are a common setup within high-throughput screening such as microtiter wellplates [11], Micropillar Microwell Array Chips (MIMICs) [12], or Droplet Microarrays (DMAs) [13], [14]. Besides, grid structures can occur in other research areas such as the analysis of parking lots using satellite images [15] or transmission electron microscopy [16].

Following the arguments of Klimaj et al. [17], manually analyzing a large number of spots within images is laborious, time-consuming, expensive, and critical in terms of reproducibility. Thus, fully automated image processing pipelines are an important goal in high-throughput screenings. However, the automatic detection and spot-wise analysis of these structures are hampered by: (i) the individual characteristics of the miniaturized platforms (shape, size, and distance between spots), (ii) different data acquisition processes from the images (e.g. rotation of the grid, different illumination conditions), (iii) the presence of artifacts in the images, or (iv) sub-grid structures meaning interruptions between grid groups.

ImageJ/Fiji [18] is the commonly used open-source tool for image analysis in biochemistry and biology. Spots can be cropped manually and processing functions such as calculating a median value are available for the image analysis of a selected crop. However, the automated detection of grid structures is not possible. The tools PlantCV [19] and CellProfiler [20] enable the definition of a grid via the specification of parameters like rows or columns. Though, neither the rotation of the grid is considered nor a complete and generic automated detection pipeline is provided. Pre-defined functions can be used to do spot-wise processing. Especially in complex scenarios, Deep Neural Networks (DNNs) can outperform traditional pre-defined image processing functions [21], e.g. presented in [22]–[26]. Hence, a pre-defined image processing function may not be sufficient for specific use-cases since, for instance, the quality requirements of the analysis are not met. In the case of PlantCV, a Graphical User Interface (GUI) is missing, which can impede the usage of the tool by researchers.

The authors in [27] define the grid location manually. Taking grid information into account, automated spot-wise processing can be executed. In the work of Klimaj et al. [17], a complete estimation of grids is avoided. First, coarse spot locations are selected manually. Subsequently, an object detection algorithm designed to obtain large objects is used to determine the true location of a spot. However, in the case of non-circular shapes, rotation can not be considered in this detection approach. Moreover, the required processing time of 15 minutes to detect 96 is comparatively large.

Having already extracted features of spots in the image, i.e. cells per spot or color information, the authors of [28] offer HTS-Corrector, a software package for statistical analysis of high-throughput screenings. Further, Chan et al. [29] present a tool for visualizing high-throughput experiment data. Though, there is no image processing integration in both tools.

The main challenges in related work can be summarized as: (i) there is no generic method for fully automated grid detection in high-throughput image analysis, (ii) available image processing tools often impede the design of custom processing functions or the integration of Deep Learning (DL) approaches, and (iii) there is no software tool including a GUI to perform automated grid detection combined with spot-wise processing enabling direct evaluation for biologist, chemists, or medical doctors.

In this paper, we introduce a novel generic tool referred to as Grid Screener to estimate grid parameters and extract grid elements. Taking this grid information into account, spot-wise processing can be done accordingly. A maximum level of flexibility is enabled through customization. Our proposed deep learning-based approach to obtain spot locations is flexible with regard to various grid structures in different assay setups. A designed robust parameter estimation algorithm reduces the requirements in terms of the accuracy of the

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3135709, IEEE Access

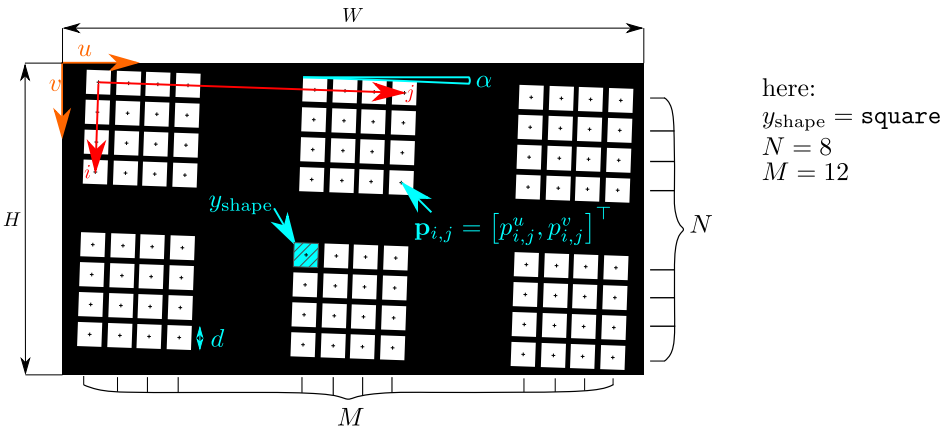


FIGURE 2: **Notation:** The centroid locations of spots i, j denoted by $\mathbf{p}_{i,j} = \left[p^u_{i,j}, p^v_{i,j}\right]^\top$, their shape y_{shape}, the corresponding expansion d, the number of rows (N), and the number of columns (M) should be determined through an automated processing. Image coordinates are denoted in the (u, v) coordinate system. The rotation of grid elements is described by a rotation angle α. Quantitative values of the illustrated exemplary parameter set are presented on the right.

DNN. Furthermore, semi-automatic grid structure detection is possible, if automated processing is not applicable, e.g. in case of no available data for training the deep neural network or variability of imaging parameters.

Our key contributions are the following:

- proposal of novel methods for the robust and flexible estimation of grid structures enabling high-throughput screening in the context of biochemical or biological experiments,
- introduction of the ready-for-use software tool Grid Screener, including a GUI, to boost the application by researchers in different scenarios, and
- demonstration of the performance benefits in the case of using our software Grid Screener considering three different datasets.

II. METHODS

A. PRELIMINARIES AND OBJECTIVES

An input image should be denoted as $\mathbf{x} \in \mathbb{N}^{H \times W}$ whereas H the image pixel height and W the image pixel width characterize (Figure 2). Image coordinates are represented in the (u, v) coordinate system. Binary pixel-wise information, whether a pixel is part of a spot or not, should be denoted by $\mathbf{y} \in \mathbb{N}^{H \times W}$. In the case of segmentation tasks, this is also often referred to as a mask. In general, estimations or predictions are represented by $\hat{(\,)}$. A DNN focusing on image segmentation is introduced as an approximate function $f_{\boldsymbol{\theta}}$ set with parameters $\boldsymbol{\theta}$ obtained during the training process of the machine learning model. Predictions of a DNN given the image $\mathbf{x}$ are noted as $\hat{\mathbf{y}} = f_{\boldsymbol{\theta}}(\mathbf{x}) \in \mathbb{N}^{H \times W}$. It applies that pixel-wise information of $\hat{\mathbf{y}}(u, v) \in \{0, 1\}$ due to the usage of the sigmoid function and subsequent binarization considering a threshold of $\frac{1}{2}$. Grid rotation angles are denoted by $\alpha \in \left(-\frac{\pi}{4}, \frac{\pi}{4}\right)$. On the one hand, the locations of spots i, j denoted by $\mathbf{p}_{i,j} = \left[p^u_{i,j}, p^v_{i,j}\right]^\top$ can be summarized comfortably in the grid tensor $\mathbf{P} \in \mathbb{N}^{N \times M \times 2}$ composed of $N \in \mathbb{Z}^+$ rows and $M \in \mathbb{Z}^+$ columns, respectively. Both are target parameters of the grid estimation. An orthogonal grid is assumed, skewed grids are out of the scope in this article. On the other hand, the shape y_{shape} of a spot is of relevance w.r.t. automated processing to select the area of interest for each spot. Hence, a crop of the spot can be extracted for all elements of the grid. In the following, we restrict our methods to the shapes of $y_{\text{shape}} \in \{\texttt{circular}, \texttt{square}\}$. Moreover, the expansion of a spot denoted as $d \in \mathbb{N}$ needs to be determined. The parameter describes the edge length in the case of square shape or the diameter for circular shape, respectively. As depicted in Figure 2, groups of spots can occur. Hence, distances between elements do not necessarily have to be equal. The case of non-equal distances between all spots is referred to as sub-grid structures.

B. IMAGE PROCESSING PIPELINE GRID SCREENER

Figure 3 presents the novel image processing pipeline of Grid Screener. Considering the entire processing pipeline of high-throughput screenings, obtaining grid parameters is a necessary previous processing step when coping with grid-shaped images. The subsequent spot-wise processing can be various depending on the underlying experiment as already presented in Figure 1. All given modules of Grid Screener are discussed in detail below.

1) Pre-processing

First, the input image $\mathbf{x}_0$ is pre-processed to obtain $\mathbf{x}$. The image is transformed to gray-level space to enable both, processing of color and gray-level images. An image normalization (mean value equals zero, standard deviation equals one) is done to ensure a proper input for the following spot detection via the DNN $f_{\boldsymbol{\theta}}$. Dealing with high-resolution images (e.g. $\geq$ 100 megapixels), down-sampling boosts the computation. Moreover, there are options of using overlapping sliding windows in the case of limitations w.r.t.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3135709, IEEE Access

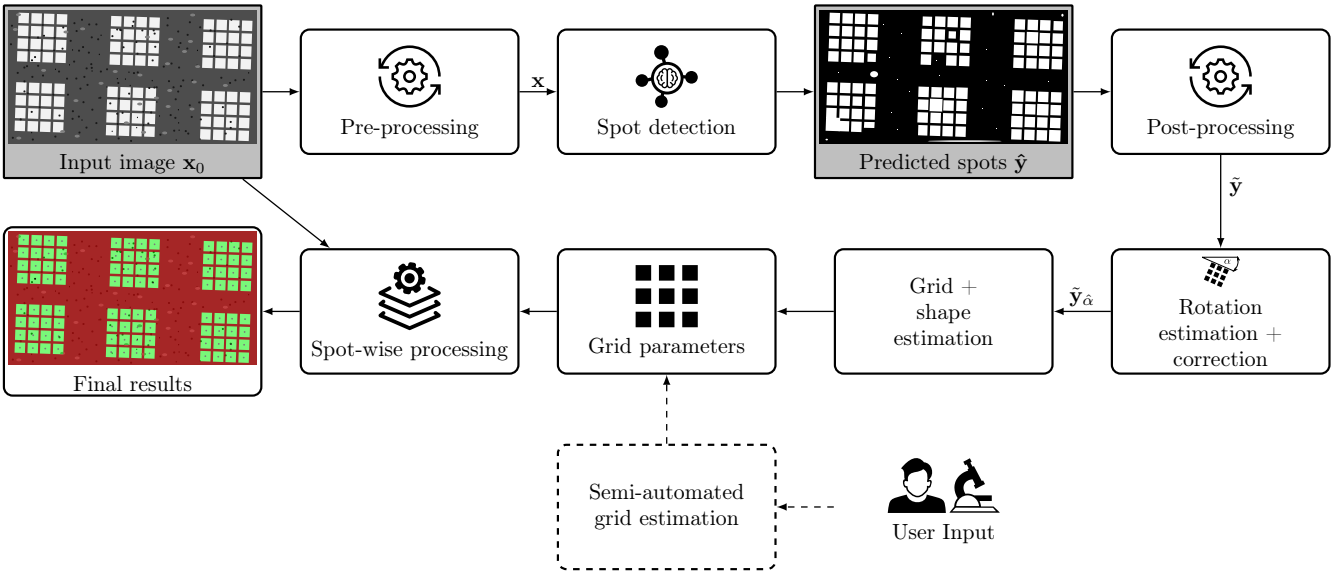


FIGURE 3: **Image processing pipeline Grid Screener:** An input image $\mathbf{x}_0$ is pre-processed yielding $\mathbf{x}$ to perform spot detection that results a prediction $\hat{\mathbf{y}}$. The post-processed predictions $\tilde{\mathbf{y}}$ are used to obtain the rotation-corrected predictions $\tilde{\mathbf{y}}_{\hat{\alpha}}$. Hence, all grid parameters can be obtained using the $\tilde{\mathbf{y}}_{\hat{\alpha}}$. The input image $\mathbf{x}$ can be rotated analogously to $\tilde{\mathbf{y}}$ enabling a spot-wise processing in the same coordinate system in order to obtain final results of grid-shaped high-throughput experiments. In addition, elements of semi-automated grid estimation are presented (dashed).

Graphics Processing Unit (GPU) memory. Thus, less powerful GPUs concerning the available memory can be used in our approach without any problems. The overlapping sliding window predictions are merged, subsequently. In addition, less accelerated processing by using CPU is possible, which yields flexibility for users of Grid Screener.

2) Spot detection

The DNN $f_{\boldsymbol{\theta}}$ serves as a spot detector calculating a prediction of spot pixels $\hat{\mathbf{y}} = f_{\boldsymbol{\theta}}(\mathbf{x})$ given image $\mathbf{x}$. Compared to traditional computer vision approaches such as hough circle/line detection, the consideration of DNNs is a more generic approach. Though, this prediction is not correct in general. For instance, damaged slides, corrupted spots, or low-expressing spots are issues where partly wrong predictions arise. Hence, post-processing is needed to enhance robustness.

3) Post-processing

The post-processing cleans the prediction $\hat{\mathbf{y}}$. First, morphological operators suppress small noisy segments and smooth shapes within the prediction. Then, all segments are filtered based on their corresponding areas. The filter limits are obtained via analyzing histogram w.r.t. the area of found objects to ensure robustness. Filtering means that segments outside the filter limits are deleted. The post-processed and filtered prediction is denoted by $\tilde{\mathbf{y}}$.

4) Rotation estimation and correction

A straightforward estimation of the grid rotation in the given image is not feasible. Hence, we propose an algorithm for a robust estimate of the rotation angle $\hat{\alpha}$. Further, we obtain the distance between two neighboring spots denoted as $\hat{r}$. A detailed description of our algorithm in pseudocode is given in the Supporting Information. Hough-based approaches to estimate rotation angle are not considered. Reasons are the required high computing time to achieve the necessary accurate angular discretization in high-resolution images and susceptibility to errors.

We assume that the number of neighboring elements in an equal-distanced sub-grid dominates the number of sub-grids. Thus, a center spot is arranged with roughly equal distances to neighboring spots as well as perpendicularity. An illustration of this assumption is given in Figure 4a. This assumption holds regarding common biochemical or biological assay platforms.

The pixel-wise spot information of predicted segments in terms of $\tilde{\mathbf{y}}$ is interpreted and centroid locations of all detected spots are obtained. Index i is used in this case for a general detection, whereas index j corresponds to neighbors of detection i. We determine the four nearest neighbors per alleged detection (u_i, v_i) using the k-nearest neighbors algorithm. These neighbors are transferred to a polar coordinate system using the alleged detection as an origin which yields magnitudes r_j and arguments φ_j to specify the neighborhood.

The polar representation can be used to execute feasibility checks. Thereby, two criteria have to be fulfilled: (i) similarity w.r.t. magnitudes of all neighbors and (ii) an angular difference between neighbors of approximately $\frac{\pi}{2}$ which can be interpreted as perpendicularity. The tolerances of accepted spread in magnitudes or arguments are selectable parameters.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3135709, IEEE Access

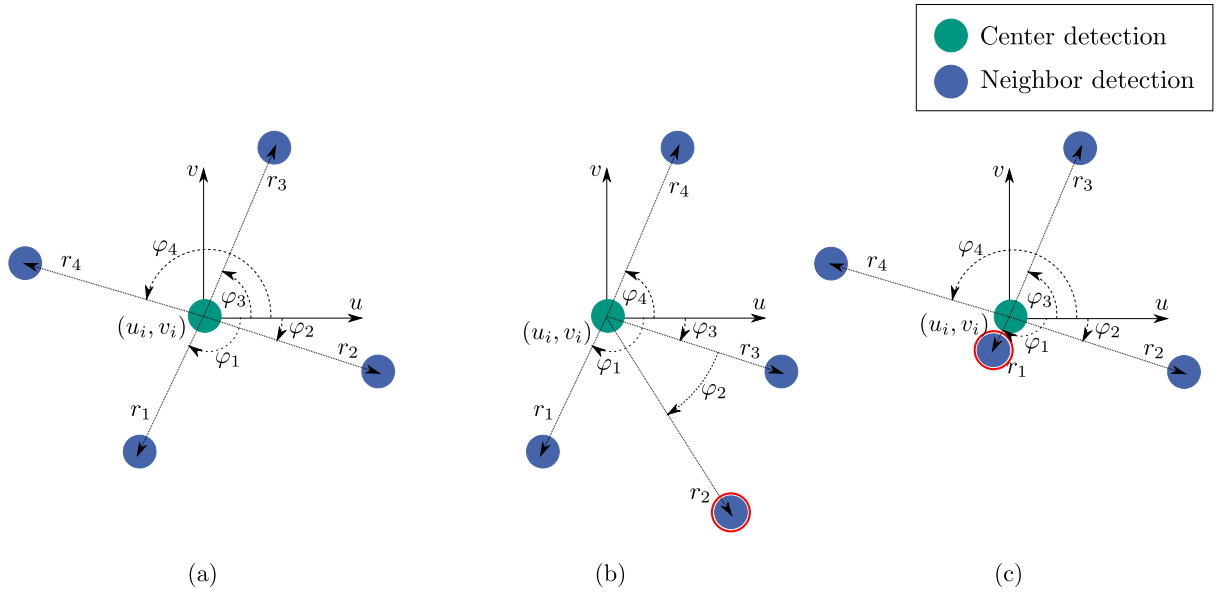


FIGURE 4: **Rotation estimation:** The center detection (u_i, v_i) is depicted in a relative polar coordinate system including magnitudes r_j and arguments φ_j of neighbors. Case (a) represents a feasible neighborhood. In contrast, infeasible neighborhoods due to violations (marked with red) w.r.t. arguments (b) and magnitude (c) are opposed. In particular, the tolerance of accepted spread in magnitudes (b) or arguments (c) are selectable parameters of the algorithm.

Figure 4b presents a non-feasible neighborhood regarding angular criterion (ii). In comparison, a violation of similar distances (i) is given in Figure 4c.

Only feasible neighborhoods are considered for further calculations. The rotation angle $\hat{\alpha}$ and spot distance $\hat{r}$ are obtained robustly according to the given intermediate steps (cf. pseudocode in the Supporting Information).

To simplify the following grid estimation, all detections are compensated by the estimated angle $\hat{\alpha}$ to obtain $\tilde{\mathbf{y}}_{\hat{\alpha}}$. Compensation means a rotation by the inverted rotation angle $\hat{\alpha}$. Consequently, this compensation yields a parallel oriented grid to the height and width dimension of the image.

5) Grid and Shape Estimation

The objective of grid estimation is to obtain the number of rows $\hat{N}$, columns $\hat{M}$, and all locations of spots summarized in $\hat{\mathbf{P}}$. A comprehensive presentation of our algorithm in the form of pseudocode can be found in the Supporting Information. The pixel-wise and rotation-corrected spot information in terms of $\tilde{\mathbf{y}}_{\hat{\alpha}}$ is transformed to a centroid detection list $\mathbf{L}$. Moreover, the lists $\mathbf{L}_u$ and $\mathbf{L}_v$ are subsets of $\mathbf{L}$ that only include u or v centroid coordinates of the detections.

All detections are clustered regarding the introduced location feature lists $\mathbf{L}_u$ and $\mathbf{L}_v$ by a clustering algorithm. Though, the clustering approach needs to be able to determine the number of clusters since this parameter is initially unknown. The number of clusters should be equal to the number of rows $\hat{N}$ or the number of columns $\hat{M}$. However, the detection list $\mathbf{L}$ may include noisy or wrong detections due to remaining errors in the segmentation $\tilde{\mathbf{y}}_{\alpha}$. Thus, the clustering algorithm needs to be robust w.r.t. noise and outliers. Thereby, the previously calculated median spot distance $\hat{r}$ helps to decide whether a new cluster for separation is required or not. The obtained cluster assignments are denoted by lists $\mathbf{C}_u$ and $\mathbf{C}_v$.

Processing the information represented in $\mathbf{C}_u$ and $\mathbf{C}_v$, each detection (u_i, v_i) is assigned to a row and to a column cluster or marked as an outlier, respectively. Hence, wrong detections can be filtered using the results of the clustering. Further, the number of clusters is assigned to the previously unknown grid parameters $\hat{N}$ and $\hat{M}$. Thereafter, a line estimate is done for each horizontal and vertical cluster separately using a parameter estimation approach. All detections that are marked as outliers by the clustering algorithm are not taken into account. The estimated line parameters, which are the offset and the slope, are stored in the lists $\mathbf{F}_u$ (vertical lines) and $\mathbf{F}_v$ (horizontal lines). A subsequent intersection calculation considers all combinations of vertical and horizontal lines. The corresponding linear equation system is solved for all $\hat{N} \cdot \hat{M}$ combinations of elements in $\mathbf{F}_u$ and $\mathbf{F}_v$. Thus, the locations of all spot centroids $\hat{\mathbf{P}}$ are determined. Thereby, the coordinates of centroid spots are rounded to integers in order to obtain pixel coordinates. Using the proposed intersection approach enables sub-grid detection directly since no assumption of equal spot distances is necessary.

The expansion of spots $\hat{d}$ is calculated robustly in terms of the median expansion of all remaining detections in $\tilde{\mathbf{y}}_{\alpha}$.

Further, shape estimation is proceeded considering the area as a feature for discrimination. The area of each detection A_i is compared to a corresponding squared $A_{\text{square},i}$ or circular $A_{\text{circle},i}$ area. The used expansion for area calculation is the

average value of horizontal and vertical expansion of each detection. To summarize the proceeding, our used classifier can be described by

$$y_{\text{shape},i} = \begin{cases} \texttt{circle}, & \|A_i - A_{\text{circle},i}\| < \|A_i - A_{\text{square},i}\| \\ \texttt{square}, & \text{else} \end{cases} \tag{1}$$

and predicts the shape for each detection i. The final shape $\hat{y}_{\text{shape}}$ is the most frequent element of all predictions $\hat{y}_{\text{shape},i}$. To estimate further or more complex shapes, the classifier can be replaced with a more elaborate approach.

6) Semi-automated Grid Estimation

In the case of failure regarding fully automated grid detection, i.e., being faced with no available training data or problematic imaging conditions, we enable semi-automated grid estimation, additionally. First, a user can define the number of rows and columns as well as the spot shape. Subsequently, selecting the corners of the grid allows obtaining rotation angle $\hat{\alpha}$ and the locations of centroids $\hat{\mathbf{P}}$. Thereby, basic trigonometric and geometric relationships are taken into account.

7) Spot-wise processing

Taking the obtained grid parameter into account, spot-wise processing is enabled. As already motivated, the processing depends on the underlying problems and thus can be customized.

C. ROBUSTNESS AND LIMITATIONS

To further examine the robustness and limitations of the Grid Screener, we consider an additional test procedure. The quality of the DNN prediction is the main influencing factor regarding a successful estimation of grids. Thereby, we change or corrupt the DNN predictions $\hat{\mathbf{y}}$. We distinguish between the synthesized corruptions or changes:

- Noise - adding random salt and pepper noise to the predicted segmentation mask,
- Missing segments - segments are deleted randomly in total,
- Additional segments - segments are added randomly,
- Rotation - the entire segmentation mask is rotated, and
- Sub-grid structure - entire rows or columns are deleted to generate a segmentation mask composed of sub-grids.

Subsequently, the grid detection performance is analyzed in the case of changed input predictions.

D. EVALUATION

1) Quantitative Metrics

To evaluate Grid Screener, we compare the time needed for users t_{user} to obtain elements of the grid. In addition, t_{process} describes the processing time including computational effort.

Further, we consider the quality of grid estimation. Considering a ground truth grid composed of detections $\mathbf{p}_{i,j}$ and the estimated grid centroids $\hat{\mathbf{p}} \in \hat{\mathbf{P}}$, the grid centroid estimation error score

$$\Delta_{\mathbf{p}} = \frac{1}{N \cdot M \cdot r} \sum_{i=1}^{N} \sum_{j=1}^{M} \underset{\hat{\mathbf{p}} \in \hat{\mathbf{P}}}{\operatorname{argmin}} \|\mathbf{p}_{i,j} - \hat{\mathbf{p}}\|_2 \tag{2}$$

describes a metric of the error during grid estimation. The metric $\Delta_{\mathbf{p}}$ depicted in Equation (2) is normalized by the present spot distance $r > 0$ between centroids of two neighboring spots within a (sub-) grid. In contrast, the grid expansion error

$$\Delta_d = \frac{\left| d - \hat{d} \right|}{d} \tag{3}$$

characterizes the relative error of the estimated expansion $\hat{d}$ of spots compared to ground truth expansion $d > 0$.

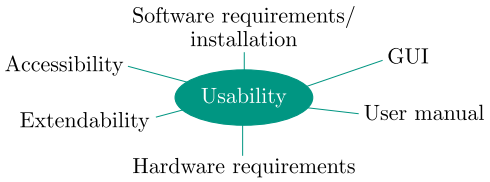


FIGURE 5: **Criteria usability study:** To evaluate the usability of the tool Grid Screener, the criteria of accessibility, software requirements/installation, available GUI, available user manual, hardware requirements, and extendability are considered.

2) Usability

In preparation for a usability study concerning Grid Screener, we conduct evaluation criteria denoted in Figure 5. On the one hand, hurdle-free access to a tool plays an important role in terms of usability. On the other hand, a GUI and a user manual enhances the usage of tools. Software and hardware requirements affect how a software tool can be used in practical projects. Moreover, the opportunity to extend an available tool can be beneficial to apply it on neighboring problem areas.

III. RESULTS

A. IMPLEMENTATION

The proposed tool Grid Screener is implemented in python. We deploy a `pip` package to enable comfortable cross-platform usage. Further, a provided user manual enables the easy and smart usage of the software for researchers. We tested the software package on Windows 10 and Ubuntu 20.04 in combination with python 3.8.5. The tool is available as public repository under https://git.scc.kit.edu/sc1357/grid-screener.

1) Deep Learning

We use the state-of-the-art convolutional neural network U-Net presented in [26] for the DNN $f_{\boldsymbol{\theta}}$ in our proposal. It is composed of a traditional autoencoder (encoder-decoder)

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3135709, IEEE Access

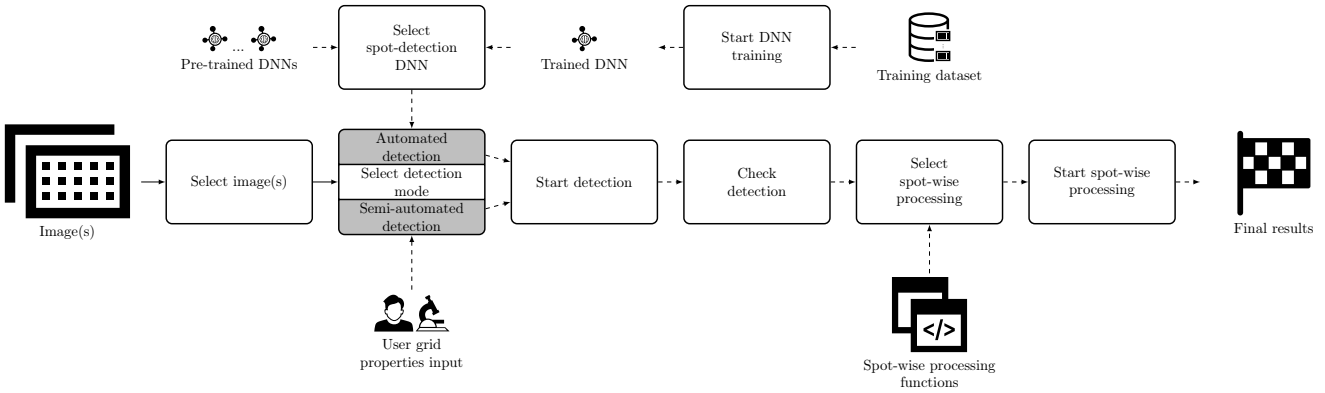


FIGURE 6: **Flow chart Grid Screener:** A single image or overlaying images must be selected. It can be chosen to do spot detection either in an automated or semi-automated manner. The semi-automated grid detection requires the grid properties as user input. In contrast, an automated detection needs a selection of the DNN used to detect spots. On the one hand, pre-trained models can be utilized. On the other hand, a custom model can be trained using an annotated training dataset. Afterwards, the detection algorithm can be started. A check of the detection result is possible before the spot-wise processing starts. Previously, the required spot-wise processing function needs to be selected. Hence, final results can be obtained.

architecture which is extended by skip connections to enhance location information. Each encoder or decoder consists of convolution, batch normalization, and rectified linear unit blocks. Dice loss [30] serves as objective function to optimize the network parameters. The architecture is able to handle small-scale data scenarios. We implement the U-Net in PyTorch Lightning [31] and use data augmentation such as image flipping, shifting, rotating, rescaling, cropping, Gaussian noise superposition, contrast adjustment, and adaptation of brightness provided by Albumentations [32]. To reduce DNN training duration, the proposed hybrid high-performance computing/high-throughput computing concept in [33] is considered. DNN training is performed on cluster nodes equipped with Intel Xeon Platinum 8368 CPU (2 sockets, 76 cores per socket) respectively NVIDIA A100 Tensor Core GPUs. Logging for interpretation of results is performed by Weights&Biases [34]. Details w.r.t. data augmentation, DNN architecture, or training are given in the Supporting Information.

2) Image Processing

Besides the DL part in the Grid Screener proposal, the image processing libraries OpenCV [35] and scikit-image [36] are used for pre-processing, post-processing, and spot-wise processing. Using a python-based implementation allows the seamless integration of other open-source python libraries. In particular, this can be useful for spot-wise processing. For instance, processing algorithms implemented in state-of-the-art DL frameworks such as PyTorch [37] can be easily used.

3) Robust Parameter Estimation

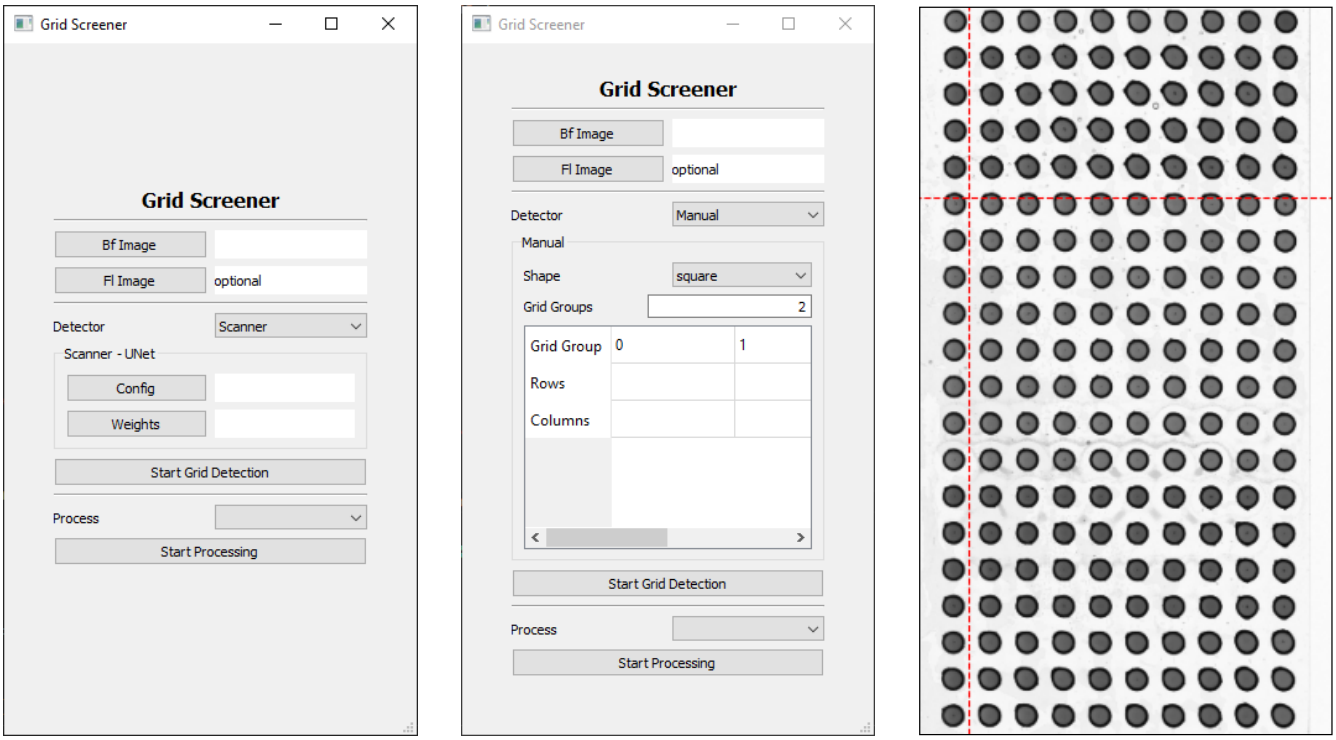
We propose to use the Density-based spatial clustering of applications with noise (DBSCAN) [38] as the clustering algorithm. Reasons for DBSCAN are robustness regarding noise as well as the non-parametric approach meaning that the number of clusters is not required as a parameter. Line estimation can be done by minimizing the sum of least squares to obtain a polynomial of order one. However, this method tends to be vulnerable regarding outliers. Thus, we propose using Random sample consensus (RANSAC) [39] to gain more robustness in our proposed algorithm. Coping with a large number of outliers represented in the dataset, RANSAC is still able to estimate model parameters accurately comparing the number of inliers and outliers of an estimated model. We use the implementation of both algorithms and k-nearest neighbors given in the library scikit-learn [40]. The library NumPy [41] is used for general calculations such as solving linear equation systems.

4) Flow Chart and GUI

To present the whole process of our proposal, a flow chart is given in Figure 6. The selection of a single image or overlaying images form the initial step in the procedure. The user can choose between automated or semi-automated spot detection. The automated detection requires a selection of the DNN used for spot detection. Thereby, considering pre-trained DNNs or training a new DNN is possible. In the case of semi-automated grid detection, grid properties need to be defined by user input. After starting and finishing the detection, the results of spot detection can be checked. Final results are obtained when the spot-wise processing has finished. However, a spot-wise processing function needs to be chosen previously.

A GUI given in Figure 7 is developed to enhance the usability for researchers. The GUI is developed using Qt5 which allows easy customization. Figure 7a presents the GUI in the case of standard fully automated processing. In contrast, Figure 7b visualizes the introduced case of semi-automated processing including input selection by users (cf. Figure 7c). The GUI integrates both, grid detection and

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3135709, IEEE Access



(a) Automated Detection (b) Semi-Automated Detection (c) User Interface Semi-Automated Detection

FIGURE 7: **GUI:** The user can select between the fully automated (a) or the semi-automated (b) grid detection approach using Grid Screener. An exemplary image concerning the interface for semi-automated grid detection is presented in (c). Grid detection can be done by using a crosshair (red intersection of lines) for user input. In the case of fully automated processing, the configuration and parameters of the DNN are required. In contrast, a few grid parameters need to be set by the user considering semi-automated grid detection. Different shapes per sub-grid are supported. In addition, spot-wise processing can be done by selecting the corresponding function using a dropdown widget "Process". Using different but overlaying images for the estimation of grid parameters and spot-wise processing is possible. A bright-field image can serve for grid detection whereas spot-wise processing is based on the associated fluorescent image. Videos of exemplary usage are given in the Supporting Information.

subsequent spot-wise processing. The configuration and parameters of the trained DNN need to be selected when using the fully automated approach. In the case of semi-automated grid detection, as presented in the method section, a part of the grid parameters needs to be set by the user. An option to use different but overlaying images for the detection and spot-wise processing is provided. For instance, the bright-field image can be used for grid estimation and the associated fluorescent image may be considered for spot-wise processing.

B. DATASETS

To investigate general applicability, we consider three types of biological or biochemical image data. Exemplary crops of input image $\mathbf{x}$ and associated mask $\mathbf{y}$ are given in Figure 8. DMA data [42] represented in square (cf. Figure 8a) and circular (cf. Figure 8b) shape are presented. Since the two types of DMA image data differ only regarding the shape, the datasets are aggregated to one dataset that is composed of 411 training and 103 test samples with an image size of 256px $\times$ 256px. Crops of the high-resolution original image are created to cope with GPU memory restrictions.

Further, we investigate our proposed Grid Screener w.r.t. common wellplates in a square shape. In contrast, the wellplate dataset presented in Figure 8c has a lower amount of training examples compared to DMA dataset. Using this small-scale dataset, we examine functionality in scenarios with less available data. The dataset includes 51 training and 13 test samples with the image size of 256px $\times$ 256px. We annotated both datasets using an in-house developed annotation tool.

C. SPOT SEGMENTATION

We evaluate the spot segmentation using the Dice-Sørensen coefficient DSC [30]. Taking the test datasets into account, average performance $DSC_{\text{test}} = 97.63\ \%$ in the case of DMA and $DSC_{\text{test}} = 96.59\ \%$ for the wellplate dataset show the capability of DNNs for spot detection. The lower performance score in the case of wellplate can be explained due to a smaller dataset as well as a more complex im-

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3135709, IEEE Access

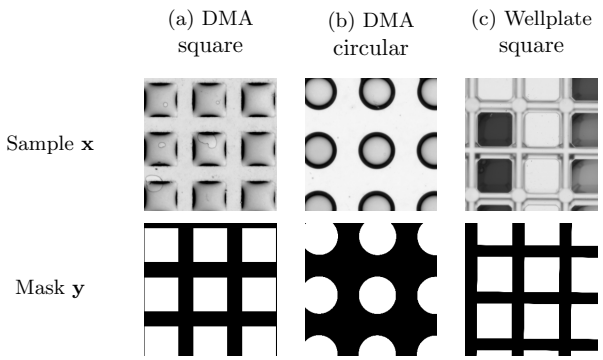


FIGURE 8: **Datasets:** Exemplary crops of images $\mathbf{x}$ and associated masks $\mathbf{y}$ are presented for the DMA data of different shapes (a) or (b) as well as wellplate data (c).

TABLE 1: **Quantitative benchmark analysis:** The needed user time effort t_{user}, processing time t_{process}, and accuracy metrics ($\Delta_{\mathbf{p}}$, Δ_d) are compared in the case of a DMA slide composed of 672 spots. Thereby, manual grid detection is opposed to Grid Screener in semi-automated and automated detection mode.

	Manual	**Grid Screener**	
		semi-automated	automated
t_{user} in s	835	61	< 0.01
$\Delta_{\mathbf{p}}$	0.01	0.02	0.01
Δ_d	0.02	0.02	0.03
t_{process} in s	835	61.01	3.88[1]/35.99[2]

[1] GPU
[2] CPU

age processing problem. For example, shadows are present within the wellplate images (cf. Figure 8c) and may impede image recognition. Though, the DNN is capable to serve as a generic detector indicated by $DSC_{\text{test}} > 90\ \%$. It can be used to predict spots of different shapes only by annotating a small dataset. The small-scale data scenario (wellplate dataset) is solved with sufficient performance. In contrast, traditional computer vision techniques like hough circle detection or hough line detection [35], [36] are designed for special shapes and thus less generic. Moreover, the manual parametrization of these methods is often burdensome.

Inference time of the DNN scales with the corresponding image size. For instance, inference time using a consumer CPU (Intel Core i7-10750H) is 34.52 s, whereas utilizing a consumer GPU (NVIDIA Quadro RTX 3000) yields an inference time of 2.41 s in the case of a DMA slide composed of 672 spots (6379px $\times$ 18992px). Hence, the application does not require a GPU. However, the usage of GPU hardware accelerates the processing by more than a factor of 10 which makes it more comfortable in the application.

D. GRID ESTIMATION

1) Quantitative Benchmark Analysis to State of the Art

To examine Grid Screener, we compare the time t_{user} needed by a user, total processing time t_{process}, and accuracy metrics $\Delta_{\mathbf{p}}$ presented in Equation (2) or Δ_d given in Equation (3). We oppose the methods of manual grid detection, semi-automated grid detection, and fully automated detection. We take $n = 10$ samples of user selections to provide a robust metric. Users are instructed by a supplied user manual. A DMA slide with 672 spots is taken under consideration to do the benchmark. To keep the experiment feasible in the case of manual detection, we measured the average time needed to mark and extract ten single spots and extrapolated them to 672 detections. Further, the accuracy of grid estimation in terms of $\Delta_{\mathbf{p}}$ and Δ_d is calculated using a subset of those ten spots.

The results are presented in Table 1. In the case of manual grid detection, $t_{\text{user}} = 835$ s is needed for obtaining the grid spots. Due to no additional computing in this case, the total processing time t_{process} is equal to t_{user}. In contrast, a semi-automated processing achieves $t_{\text{user}} = 61$ s and is superior to the manual grid detection approach regarding needed time $t_{\text{user}}/t_{\text{process}}$. Comparing t_{user} and t_{process} in this case, the computational effort (0.01 s) is negligibly small. However, our fully automated approach requires no user input to estimate the grid ($t_{\text{user}} < 0.01s$). The processing time of Grid Screener is composed of DNN inference time to generate an image segmentation and grid estimation computing time, respectively. As previously mentioned, DNN inference time differs depending on the used hardware device. Total processing time for grid estimation in the case of an available GPU (NVIDIA Quadro RTX 3000) of $t_{\text{user}} = 3.88$ s is superior to CPU processing time of $t_{\text{user}} = 35.99$ s. Though, taking quantitative metrics $t_{\text{user}}/t_{\text{process}}$ into consideration, Grid Screener is superior in all cases to manual processing.

Taking the accuracy metrics $\Delta_{\mathbf{p}}$ and Δ_d into account, all methods show no outstanding differences (cf. Table 1).

Hence, Grid Screener is the overall most performing approach. The reduction of needed user time whilst keeping accuracy on a high level ($\Delta_{\mathbf{p}} \ll 0.1, \Delta_d \ll 0.1$) is a major advantage within practical experiments and makes an important contribution to improve high-throughput screening.

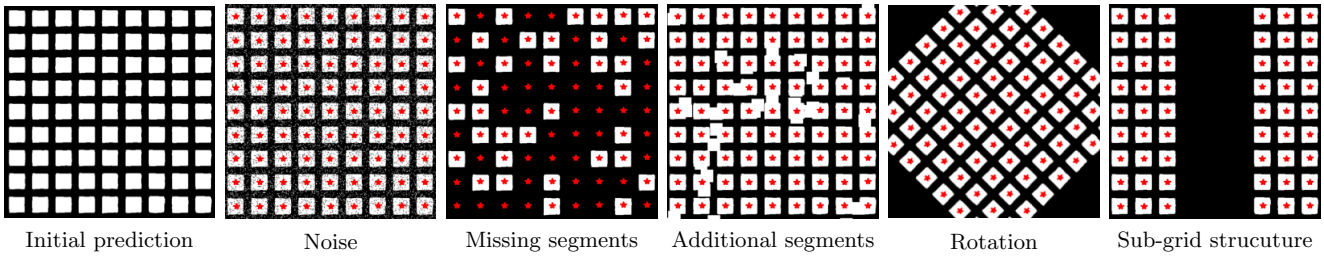
2) Accuracy Evaluation on Test Assays

To avoid overfitting in our proposed grid estimation algorithms, we test Grid Screener on two complete slides per dataset in contrast to the previously considered crops of slides. Thereby, we consider the introduced normalized metrics in Equation (2) and Equation (3). The introduced grid parameters could be estimated with sufficient accuracy in all test cases indicated by resulting metrics $\Delta_{\mathbf{p}} \leq 0.053$ and $\Delta_d \leq 0.037$. Further, y_{shape} is classified in all test slides with 100% accuracy. For more detailed results such as the obtained metrics per test image or the resulting grid estimation per test case refer to the Supporting Information.

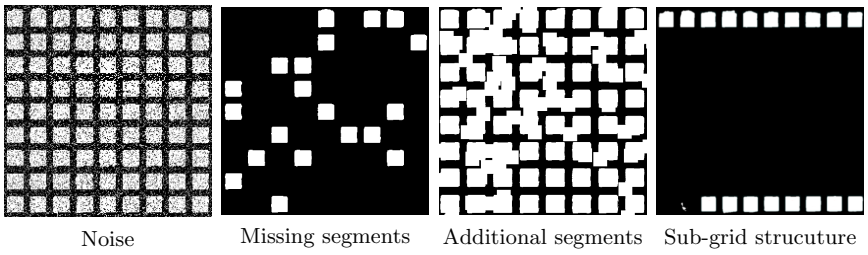
3) Robustness and Limitations

An excerpt of the results is depicted in Figure 9. A failure case is defined by $\Delta_{\mathbf{p}} \geq 0.1$ or $\Delta_d \geq 0.1$. Figure 9a shows cases Grid Screener is able to cope with. In contrast,

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3135709, IEEE Access



(a) **Robustness:** The robustness of the grid mask detection is tested. Thereby, the initial prediction is corrupted with salt and pepper noise, segments are deleted or added, the entire segmentation mask is rotated, and sub-grid structures are generated. Red crosses represent the centroids of the estimation performed by Grid Screener.



(b) **Limitations:** A failure case is defined by $\Delta_{\mathbf{p}} \geq 0.1$ or $\Delta_d \geq 0.1$. Taking a salt and pepper noise level of 30%, 70% missing segments, 80% additional segments, or a sub-grid in combinations with missed detections into account, Grid Screener fails to detect the grid with sufficient accuracy.

FIGURE 9: **Robustness and limitations to imperfect spot predictions:** Robustness (a) and limitations (b) are compared concerning different corruptions.

Figure 9b presents instances in which the limitations of Grid Screener are visualized. A salt and pepper noise level of 30% leads to failure. However, such a large noise level is unlikely when dealing with convolutional neural networks since convolution kernels smooth predictions. Further, 70% of missing segments or a sub-grid in combinations with missed detections lead to problems since only a single representative of row and columns will be classified as noise by DBSCAN. Besides, 80% of additional segments leads to a merging of segments. Consequently, rows and columns cannot be discriminated using Grid Screener which leads to a failure. Though, rotations of the initial segmentation mask lead to no failure during the processing.

To sum up the analysis in general, Grid Screener is robust to an amount of corruption in which the overall grid structure remains visible. A detailed overview concerning the results of the robustness and limitations analysis is given in the Supporting Information.

4) Spot-wise Processing

Exemplary spot-wise processing is depicted in Figure 10. The results of Grid Screener are used to perform spot extraction and save each spot in separate images utilizing the "crop spot" function.

E. USABILITY STUDY

The results of the usability study are presented in Table 2. By providing a public code repository including a user manual in form of a `README` file, accessibility and user manual can be marked as full target achievement. Grid Screener can be used via the provided GUI. Further, deploying the software in python as `pip` package, software requirements are low and installation can be done by users comfortably. However, to generate results in less computing time, a GPU is beneficial. Though, Grid Screener offers a CPU mode leading to increased computing time. Hence, the aspect of low hardware requirements is not fulfilled completely. Grid Screener can be extended by other spot detectors or additional spot-wise processing functions. The user needs to be able to write a custom python function. Thus, there is only partial target achievement in terms of extendability.

TABLE 2: **Usability study:** The introduced criteria are evaluated using ✓ to indicate full, (✓) for partial, and an empty cell for no target achievement.

Criterion	Evaluation
Accessibility	✓
User manual	✓
GUI	✓
Software requirements/installation	✓
Hardware requirements	(✓)
Extendabilty	(✓)

Therefore, we can demonstrate the usability of Grid Screener for researchers by fully satisfying four criteria and partially satisfying two criteria.

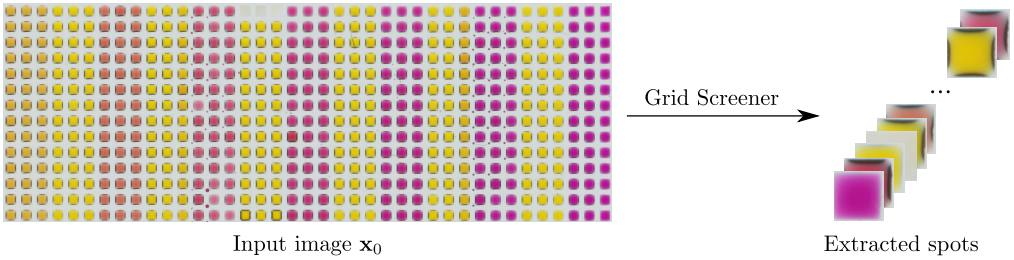


FIGURE 10: **Spot-wise processing:** All spots given in the input image $\mathbf{x}_0$ are detected by Grid Screener. Subsequently, each spot is extracted and saved using the implemented "crop spot" function.

IV. CONCLUSIONS

High-throughput assays of biochemical or biological experiments often consider grid structures of spot arrays. Thereby, image processing is a common method to perform analysis referred to as high-throughput screening. However, there are no generic methods coupled with software packages for the automated analysis of grid-shaped images available. Hence, the often considered manual processing leads to additional time effort, high costs, and low reproducibility of the processing for researchers during image analysis.

We introduce Grid Screener which is a generic tool for detecting grid structures and subsequent spot-wise processing in biochemical or biological images. To enable user-friendly operation for researchers such as biologists, chemists, or medical doctors, we provide a developed software package including graphical user interface with a corresponding manual. The novel tool combines a deep learning-based approach and a robust parameter estimation algorithm to obtain generic grid structures. The automated processing guarantees a reproducible experiment evaluation, which helps researchers when comparing different settings. Custom spot-wise processing allows the usage of Grid Screener in a wide range of applications such as colorimetric analysis, cell detection, or segmentation of spheroids.

Three different grid structures with different shapes are considered to evaluate the tool in practical applications. First, Grid Screener demonstrates the benefits regarding the needed time for users to do image analysis. On the one hand, the processing time is reduced. On the other hand, the effort is transferred from researchers to computers. Second, we show the robustness of our proposal w.r.t. synthesized changes or inserted corruptions into predictions of the deep neural network. Hence, this results in lower accuracy requirements in terms of estimating grid segments using deep learning.

Grid Screener is not only suitable for high-throughput screening, an analysis of grid structures is relevant during the manufacturing of slides for automated quality control or can be integrated into experimental robot systems dealing with grid structures.

Part of future work is the extension and integration of further commonly used spot-wise image processing functions. In particular, individual spot-wise processing algorithms can be made available for the community using the established code repository. Moreover, investigations to reduce the computing time of the deep neural network, especially in the case of no available GPU, for further improvement are pending. Further, we are investigating neighboring problems such as the analysis of parking lots by means of satellite images and are undertaking proof-of-concept experiments. For instance, the initial working package here is the creation of datasets corresponding to other research domains. In addition, object detection in other domains brings further challenges such as coping with low-resolution satellite images.

Grid Screener as an open-source tool can contribute considerably to the research community and can help to make a further step concerning lab automation in the context of high-throughput screening.

REFERENCES

[1] C. Gallert, R. Lehmann, T. Roddelkopf, S. Junginger, and K. Thurow, "High throughput screening system for screening of 3D cell cultures," in *2015 IEEE International Instrumentation and Measurement Technology Conference*, 2015, pp. 1302–1307.

[2] P. Szymański, M. Markowicz, and E. Mikiciuk-Olasik, "Adaptation of high-throughput screening in drug Discovery—Toxicological screening tests," *International Journal of Molecular Sciences*, vol. 13, no. 1, pp. 427–452, 2012.

[3] W. F. An and N. Tolliday, "Cell-based assays for high-throughput screening," *Molecular Biotechnology*, vol. 45, no. 2, pp. 180–186, 2010.

[4] S. Vyawahare, A. D. Griffiths, and C. A. Merten, "Miniaturization and parallelization of biological and chemical assays in microfluidic devices," *Chemistry & Biology*, vol. 17, no. 10, pp. 1052–1065, 2010.

[5] S. W. Song, S. D. Kim, J. Kim, and S. Kwon, "One-step generation of drug-releasing microarray for high-throughput screening of sequential drug combinations," in *2019 IEEE 32nd International Conference on Micro Electro Mechanical Systems*, 2019, pp. 515–518.

[6] W. Lei, K. Demir, J. Overhage, M. Grunze, T. Schwartz, and P. A. Levkin, "Droplet-microarray: Miniaturized platform for high-throughput screening of antimicrobial compounds," *Advanced Biosystems*, vol. 4, no. 10, p. 2000073, 2020.

[7] M. A. Abdul Sisak, F. Louis, I. Aoki, S. H. Lee, Y.-T. Chang, and M. Matsusaki, "A near-infrared organic fluorescent probe for broad applications for blood vessels imaging by high-throughput screening via 3D-Blood vessel models," *Small Methods*, vol. 5, no. 8, p. 2100338, 2021.

[8] T. Tronser, K. Demir, M. Reischl, M. Bastmeyer, and P. A. Levkin, "Droplet microarray: Miniaturized platform for rapid formation and high-throughput screening of embryoid bodies," *Lab on a chip*, vol. 18, no. 15, pp. 2257–2269, 2018.

[9] X. Fang, Y. Zheng, Y. Duan, Y. Liu, and W. Zhong, "Recent advances in design of fluorescence-based assays for high-throughput screening," *Analytical Chemistry*, vol. 91, no. 1, pp. 482–504, 2019.

[10] D. Baud, N. Ladkau, T. S. Moody, J. M. Ward, and H. C. Hailes, "A rapid, sensitive colorimetric assay for the high-throughput screening of transaminases in liquid or solid-phase," *Chemical Communications*, vol. 51, no. 97, pp. 17 225–17 228, 2015.

[11] D. S. P. Auld *et al.*, "Microplate Selection and Recommended Practices in High-throughput Screening and Quantitative Biology." *Assay Guidance Manual*, 2020.

[12] P. Bandaru *et al.*, "A microfabricated sandwiching assay for nanoliter and high-throughput biomarker screening," *Small*, vol. 15, no. 15, p. 1900300, 2019.

[13] Y. Liu, T. Tronser, R. Peravali, M. Reischl, and P. A. Levkin, "High-Throughput screening of cell transfection enhancers using miniaturized droplet microarrays," *Advanced biosystems*, vol. 4, no. 3, p. 1900257, 2020.

[14] A. A. Popova *et al.*, "Facile one step formation and screening of tumor spheroids using droplet-microarray platform," *Small*, vol. 15, no. 25, 2019.

[15] S. Zambanini, A.-M. Loghin, N. Pfeifer, E. M. Soley, and R. Sablatnig, "Detection of parking cars in stereo satellite images," *Remote Sensing*, vol. 12, no. 13, 2020.

[16] X. Liu and L. Gu, "Advanced transmission electron microscopy for electrode and solid-electrolyte materials in lithium-ion batteries," *Small Methods*, vol. 2, no. 8, p. 1800006, 2018.

[17] S. D. Klimaj, Y. Licon Munoz, K. Del Toro, and W. C. Hines, "A high-throughput imaging and quantification pipeline for the EVOS imaging platform," *PLoS ONE*, vol. 15, no. 8, pp. 1–13, 2020.

[18] J. Schindelin *et al.*, "Fiji: an open-source platform for biological-image analysis," *Nature Methods*, vol. 9, no. 7, pp. 676–682, 2012.

[19] M. A. Gehan *et al.*, "PlantCV v2: Image analysis software for high-throughput plant phenotyping," *PeerJ*, vol. 5, p. e4088, 2017.

[20] A. E. Carpenter *et al.*, "CellProfiler: Image analysis software for identifying and quantifying cell phenotypes," *Genome Biology*, vol. 7, no. 10, 2006.

[21] N. O. Mahony *et al.*, "Deep learning vs. Traditional computer vision," in *Advances in Computer Vision*, 2019, pp. 128–144.

[22] F. Isensee, P. F. Jaeger, S. A. A. Kohl, J. Petersen, and K. H. Maier-Hein, "nnU-Net: A self-configuring method for deep learning-based biomedical image segmentation," *Nature Methods*, vol. 18, no. 2, pp. 203–211, 2021.

[23] J. C. Caicedo *et al.*, "Nucleus segmentation across imaging experiments: The 2018 Data Science Bowl," *Nature Methods*, vol. 16, no. 12, pp. 1247–1253, 2019.

[24] V. Ulman *et al.*, "An objective comparison of cell-tracking algorithms," *Nature Methods*, vol. 14, no. 12, pp. 1141–1152, 2017.

[25] R. M. Thomas and J. John, "A review on cell detection and segmentation in microscopic images," in *2017 IEEE International Conference on Circuit, Power and Computing Technologies*, 2017, pp. 1–5.

[26] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *Medical Image Computing and Computer-Assisted Intervention*, 2015, pp. 234–241.

[27] J. W. Lim, K. S. Shin, J. Moon, S. K. Lee, and T. Kim, "A microfluidic platform for high-throughput screening of small mutant libraries," *Analytical Chemistry*, vol. 88, no. 10, pp. 5234–5242, 2016.

[28] V. Makarenkov, D. Kevorkov, P. Zentilli, A. Gagarin, N. Malo, and R. Nadon, "HTS-Corrector: Software for the statistical analysis and correction of experimental high-throughput screening data," *Bioinformatics*, vol. 22, no. 11, pp. 1408–1409, 2006.

[29] T. P. S. Chan, P. Malik, and R. Singh, "An interactive visualization-based approach for high throughput screening information management in drug discovery," in *2006 International Conference of the IEEE Engineering in Medicine and Biology Society*, 2006, pp. 5794–5797.

[30] S. Jadon, "A survey of loss functions for semantic segmentation," in *2020 IEEE International Conference on Computational Intelligence in Bioinformatics and Computational Biology*, 2020, pp. 1–7.

[31] W. Falcon, "PyTorch lightning," [Online]. Available: https://github.com/PyTorchLightning/pytorch-lightning, 2019.

[32] A. Buslaev, V. I. Iglovikov, E. Khvedchenya, A. Parinov, M. Druzhinin, and A. A. Kalinin, "Albumentations: Fast and flexible image augmentations," *Information-an International Interdisciplinary Journal*, vol. 11, no. 2, 2020.

[33] M. P. Schilling *et al.*, "A computational workflow for interdisciplinary deep learning projects utilizing bwhpc infrastructure," in *7th bwHPC-Symposium, submitted paper*, 2021.

[34] L. Biewald, "Experiment tracking with weights and biases," [Online]. Available: https://github.com/wandb, 2020.

[35] G. Bradski, "The opencv library." *Dr. Dobb's Journal: Software Tools for the Professional Programmer*, vol. 25, no. 11, pp. 120–123, 2000.

[36] S. Van der Walt *et al.*, "scikit-image: image processing in python," *PeerJ*, vol. 2, p. e453, 2014.

[37] A. Paszke *et al.*, "PyTorch: An imperative style, high-performance deep learning library," in *Advances in Neural Information Processing Systems*, 2019.

[38] 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*, E. Simoudis, J. Han, and U. M. Fayyad, Eds., 1996, pp. 226–231.

[39] M. A. Fischler and R. C. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography," *Communications of the ACM*, vol. 24, no. 6, pp. 381–395, Jun. 1981.

[40] F. Pedregosa *et al.*, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[41] C. R. Harris *et al.*, "Array programming with NumPy," *Nature*, vol. 585, no. 7825, pp. 357–362, 2020.

[42] A. A. Popova, K. Demir, T. G. Hartanto, E. Schmitt, and P. A. Levkin, "Droplet-microarray on superhydrophobic–superhydrophilic patterns for high-throughput live cell screenings," *RSC Advances*, vol. 6, no. 44, pp. 38 263–38 276, 2016.

MARCEL P. SCHILLING received the bachelor's and master's degrees in mechanical engineering from the Karlsruhe Institute of Technology, Karlsruhe, Germany, 2018 and 2020, respectively. He is a doctoral researcher in the research group "Machine Learning for High-Throughput and Mechatronics" with the Institute for Automation and Applied Informatics, Karlsruhe Institute of Technology, Karlsruhe, Germany. His research interests include image processing, data-centric deep learning, and machine learning for high-throughput screening.

SVENJA SCHMELZER received the bachelor's degree in molecular biotechnology from the Heidelberg University, Heidelberg, Germany in 2018. She is working as an intern student in the research group "Machine Learning for High-Throughput and Mechatronics" with the Institute for Automation and Applied Informatics, Karlsruhe Institute of Technology, Karlsruhe, Germany. Her research interests include biotechnology, image processing, and machine learning.

JOAQUÍN EDUARDO URRUTIA GÓMEZ received the bachelor's in biological sciences and master's degrees in bioengineering from the Universidad de Concepción, Concepción, Chile, 2017 and 2019, respectively. He is a doctoral researcher at the Institute of Biological and Chemical Systems, Karlsruhe Institute of Technology, Karlsruhe, Germany. His research is focused on developing novel in vitro systems for high-throughput screenings and personalized medicine.

ANNA A. POPOVA is a head of biological sub-group in the Multifunctional Materials Systems research laboratory at the Institute of Biological and Chemical Systems at Karlsruhe Institute of Technology. She graduated from the department of Cell Biology and Immunology of the Biological Faculty, Lomonosov Moscow State University in Russia and obtained her Ph.D. in Cell and Molecular Biology in at University of Heidelberg, Germany. Since January 2014 Dr. Popova started at KIT as a project leader and head of the biology sub-group. Her research is focused on developing novel in vitro systems for high-throughput screenings and personalized medicine.

PAVEL A. LEVKIN is head of the Multifunctional Materials Systems research group at the Institute of Biological and Chemical Systems and Institute of Organic Chemistry. He graduated from the Institute of Fine Chemical Technology, Moscow and obtained his Ph.D. in Organic Chemistry from the University of Tübingen in Germany, followed by postdoctoral work at the University of California, Berkeley. His research focuses on the development of functional and responsive materials, and surfaces for biomedical and biotechnological applications.

MARKUS REISCHL received the Dipl.-Ing. and Ph.D. degrees in mechanical engineering from the University of Karlsruhe, Karlsruhe, Germany, in 2001 and 2006, respectively. Since 2020, he is an Adjunct Professor with the Faculty of Mechanical Engineering and is heading the research group “Machine Learning for High-Throughput and Mechatronics” with the Institute for Automation and Applied Informatics, Karlsruhe Institute of Technology, Karlsruhe, Germany. His research interests include man–machine interfaces, image processing, machine learning, and data analytics.