Colonizer: Anandroid OS based automated microbial colony counter

Shashwat Kumar[1,$], Rudra Vaswata Roy Choudhury[1,2,$] and Sunil Laxman[1]

[1]Institute for Stem Cell Biology and Regenerative Medicine
NCBS campus, GKVK
Bellary Road, Bangalore 560065

[2]Dept. of Electrical Engineering
Indian Institute of Technology Madras
Chennai, Tamil Nadu 600036

[$]Authors contributed equally to the study

Email: shashwatk@ncbs.res.in, sunil@instem.res.in

Abstract:

Colony counting to estimate colony forming units (CFUs) is a standard microbiology procedure used for determining the number of viable microbes in a system. Typically this process is done manually, which is time consuming, tedious, and sometimes error prone. Alternate image analysis based systems are also time consuming. In this paper we develop and demonstrate a simpleandroid based automated colony counter, called Colonizer, also made freely available on Google Playstore. Our algorithm can also threshold and segment overlapping colonies and thus provide rapid, accurate counts across a wide range of colony densities (from 50-500 colonies/plate), for different microbes, using any android based phone or device.

Introduction:

CFUs (Colony Forming Units) are estimates of the number of viable microbes (bacteria, archaea or fungi) in a given system, and has been a microbiology staple for decades [1]. Counting of CFUs provides information on how microbial cell viability is affected by different environmental or pharmacological conditions. Typically, the estimation is performed by plating microbial cultures on nutrient-agar petridish and counting the colonies formed manually. This is a time consuming, tedious and error prone process, widely despised by most student researchers, and the tedium and error due to manual CFU estimation has not changed [2].

Efforts to automate the estimation of CFUs include using a DSLR camera or a flatbed scanner for image acquisition connected to a computer. For example, Putman et al, 2005[3] previously suggested a setup with a Canon Powershot A75 coupled with a commercial PROTOCOL3 counter for automated colony counting. Zhang et al, 2009 [4] designed an automatic bacterial colony counting system using a Nikon D50 DSLR for image acquisition and an Intel Core 2 Duo PC for running the watershed algorithm for segmenting colonies. Recently, Brugger et al, 2012 [5] came up with a hardware setup for automatic counting of bacterial CFUs on agar plates. They used a circular darkfield illuminator with an uEye UI1640C Camera for imaging and a PCfor adaptive thresholding and segmenting colonies.

These setupsrequire additional hardware components, some training or learning, and technical expertise for the design of the system. However, with available processing power present in typical smartphones, most image processing algorithms can be run on the smartphone itself. Hence we develop a novel android based software, Colonizer,combining the image acquisition and analysis steps inside the smartphone. We also suggesta simple setup to ensure uniform lighting conditions for better accuracy. Using this, no manual selection of region of interest,or tweaking of parameters is required. Simply taking the picture of the petridish is sufficient. The end result is anaccessible, portable andaccurate colony counting system for estimating CFUs across arange of colony densities (50-500 colonies/plate), for typical laboratory microbes such as *E. coli* and *S. cerevisiae*.

PeerJ Preprints | https://doi.org/10.7287/peerj.preprints.2792v1 | CC BY 4.0 Open Access | rec: 8 Feb 2017, publ: 8 Feb 2017

Results and discussion:

Before we describe the algorithm developed, we first recommend a way to take petri plate images, so as to minimize errors due to poor lighting or image quality. We used a small cardboard box (as shown in Fig 1 A-B) with a small hole to ensure uniform illumination, as described in the materials and methods. This set up reduces interference due to non-uniform lighting, and removes any noise from the edges of the petridish. Thissetup takes ~5 minutes to build with readily available components, and avoids noise due to glare, shiny colonies, phone shadows etc which can appear if the petri plate is imaged directly in ambient lighting (Fig 2 A-B). We also provided a schematic illustrating the construction steps (Fig 1C){http://imgur.com/a/Bxn6b}.

Apart from the cardboard box setup, we also tested the android app in a standard laminar-flow hood and under (variable) ambient lighting conditions. The petridish was kept vertically in the hood to ensure uniform illumination and to avoid reflections and glare. Analysis of the data for all three methods (described subsequently) suggests that images with unwanted light reflections or poor quality will reduce accuracy as shown in Fig 2 (A-B), etc when the images are clicked outside the hood under ambient light conditions.

Algorithm Overview

The image processing algorithm was written using standard OpenCV [6] library functions. The first step of this algorithm is to apply a top-hat transformation [7, 8] on the selected image to correct non-uniform illumination. Since non-uniform illumination can result in noise and artefacts leading to false counts, it is important to maintain uniformity in illumination for the image. Top-hat transform achieves this by adopting a background equalization which helps in feature extraction in an image. The image is then binarized through Otsu's thresholding [9, 10, 11]. The algorithm then detects and filters out all abnormally-sized contours and draws the remaining ones on a black background. Following this, a distance transform [12, 13, 14] for the black and the white contour image is performed. This step is necessary to segment connected contours into individual components. By finding the regional maxima of the distance-transform contours, we can estimate the centres of the colonies, whose count gives the number of colonies. These processes are illustrated in Figures3 and 4.

Thus, through this method, the algorithm cancount colonies which are connected to each other or overlapping. Execution of the different steps of the algorithm finally counts the number of colonies in the plate which is displayed on the screen below the selected image.

Algorithm features

(a) Tophat transform:

The intensity values in a gray scale image can be considered as elevations, such that the image is composed of mountain tops (bright colonies) and valley lows (dark background).Uneven illumination, which leads to uneven contrast in an image often degrades the threshold separation of the adjacent mountain peaks between a valley. To correct this and produce an image with enhanced contrast, the tophat transform is applied on an image. Fig 4A illustrates how the tophat transform works.

(b) Otsu's thresholding:

Otsu's thresholding is a binarization algorithm used in image processing [9]. It involves iterating through all the possible threshold values and classifying each pixel as either background or foreground depending on whether it's larger than the threshold. The algorithm finds the optimum threshold where the inter class variance between foreground and the background pixels are maximized. This helps producing an image with a sharp contrast difference between the background and foreground, and hence makes it possible to detect binary contours in the image. (Fig 4B)

(c) Outlier and noise removal:

This step involves removing artefacts and noise present in the images by detecting contours. We remove potentially noisy contours by filtering out the ones having an area<20pixel units.

(d) Applying distance transform and finding the regional maxima:

The distance transform is an operator applied to binary images. It transforms the input image into a graylevel image, where the graylevel intensities of points inside foreground regions are changed to show the distance to the closest boundary from each point. Following this, the regional maximum of the distance transformed image is sorted out. This helps to separate overlapping contours in the petri dish images and thereby ensures that each colony in a cluster is counted as an individual. (Fig 4C)

Accuracy

The accuracy for the Android app was determined by comparing the manual counts with the algorithm count values (Table 1). *S. cerevisiae* and *E. coli* colony images were separately analysed to determine the accuracy of the designed algorithms.

The accuracy was subject to the image acquisition method used. A mean accuracy of 91.4±5.9, 92.0±5.4, and 82.9±11.7% respectively was observed for images taken within a laminar hood, images taken using the cardboard box setup, and images taken on the bench/ambient light respectively. Images taken under ambient lighting shows substantially lower accuracy and higher fluctuations than the box and hood, as also illustrated in Fig 5A. We also tested the app performance across different devices and found low variations amongst them (Table 1).

Thus, we conclude that the most accurate count, with a low variance, is obtained by taking images inside the hood or by using the cardboard box and both of these methods are significantly better than taking images in ambient condition.

Repeatability

To ensure that the app performs consistently, we took multiple measurements of the same plate on two different phones. Low variance count on both shows that the algorithm counts do not fluctuate on repeating the measurement (Fig 6).

Table 1 illustrates the entire dataset of counting colonies by using multiple smartphone models under different image acquisition conditions.

Speed

The speed of the application depends upon the processing capacity of the phone used. We tested the algorithm on smartphones with both very low end (Micromax Q414) and higher end processors(MOTO G 4 PLUS). The execution time ranges from 20sec for the Micromax Q414 to 8sec for MOTO G 4 PLUS. It is to be noted that the Colonizer app does not require the user to manually select any parameters and the image processing algorithm is completely automated. Thus, it can be stated that significantly short time span is required to analyze the image and produce the count.

We conclude by summarizing that Colonizer is an easily usable, accurateapplication whichoperates on any ANDROID OS based device, for estimating CFUs in a range of typical laboratory microbes, from bacteria to yeasts. The application is free for download and use on Google Playstore, and the soure code is open. We hope the application is widely adopted by students and microbiological researchers.

Materials and Method:

Strains and Growth Conditions

S. cerevisiae colony growth: Yeast colonies, grown on Yeast extract, peptone, dextrose (YPD)-agar in standard 9 cm petri dishes,and grown at 30 °C were used to collect sample images. A single batch of yeast colonies of varying densities between 50 - 650 were grown on 4 agar plates.

E. coli colony growth: Bacterial colonies, grown on LB-agar in standard 9 cm petri dishes, were used as sample images. Bacterial colonies of slightly varying densities in the range of 50 - 400 were grown on agar plates using standard microbiological procedures and grown at 37 °C.

Cardboard Box Setup for correct lighting

A small cardboard box (illustrated in Fig 2 A-C) with a movable lid and a small hole was used to ensure uniform illumination. One face of the box is kept open to ambient light which acts as the light source for imaging. The petridish sits at the bottom of the box on a black sheet. The smartphone is attached to the lid such that the camera lens aligns with the hole in the lid. The lid is vertically adjusted to focus the app cropping circle on the inside boundary of the petridish (as also shown in Fig 1D). This removes any noise from the edges of the petridish. The advantage of this setup is that it could be built with readily available components and avoids noise due to glare, shiny colonies, phone shadow etc which can appear if the petri plate is imaged directly in ambient lighting (Fig 1 A-B).

Testing on different phones

We tested the app on five different smartphones running ANDROID OS, viz MICROMAX Q414, Lenovo Vibe, HTC, MOTOG, and MOTO G 4 PLUS. The camera quality ranged from 5-16megapixels, all with built-in CMOS sensors.

Acknowledgements

The authors thank Sneha Das, Vineeth V, Zeenat Rashida,V Soumya, PabitroNandi and Laasya Samhitafor providing us with microbial plates, smartphones and app validation and testing. SL is funded through a Wellcome Trust DBT-IA Intermediate Fellowship (IA/I/14/2/501523). RVRC is supported by a BERI Fellowship from NCBS/inStem/C-CAMP.

References

[1] Breed, R and WD Dotterrer. 1916. The Number of Colonies Allowable On Satisfactory Agar Plates. *J Bacteriol*. 1:321-331

[2] Jennison, MW and GP Wadsworth. 1940. Evaluation of the Errors Involved In Estimating Bacterial Numbers by the Plating Method. *J Bacteriol*. 39:389-397

[3]Putman M, Burton R, Nahm MH 2011. Simplified method to automatically count bacterial colony forming unit. *Journal of Immunological Methods Biology* 87: 1135–1146. doi: 10.3109/09553002.2011.622033

[4] Chen, WB. & Zhang2009. An automated bacterial colony counting and classification system*C. Inf Syst Front* 11: 349. doi:10.1007/s10796-009-9149-0

[5] Brugger SD, Baumberger C, Jost M, Jenni W, Brugger U, Mühlemann K 2012. Automated Counting of Bacterial Colony Forming Units on Agar Plates. *PLoS ONE* 7(3): e33695. doi:10.1371/journal.pone.0033695

[6] Guobo Xie and Wen Lu 2013. Image Edge Detection Based on OpenCV.*International Journal of Electronics and Electrical Engineering*1(2): 104-106

[7] Hamid Hassanpour, Najmeh Samadiani, S.M. Mahdi Salehi 2015. Using morphological transforms to enhance the contrast of medical images. *The Egyptian Journal of Radiology and Nuclear Medicine* 46: 481-489. doi: 10.1016/j.ejrnm.2015.01.004

[8] Suman Thapapr and Shevani Garg 2012., Study and Implementation of Various Morphology Based Image Contrast Enhancement Techniques.*Proceedings of 'I-Society 2012', International Journal of Computing and Business Research* ISSN 2229-6166

[9] Nobuyuki Otsu 1979. A threshold selection method from gray-level histograms.*IEEE Trans. Sys., Man*., Cyber. 9 (1): 62–66. doi:10.1109/TSMC.1979.4310076.

[10] Astha Baxi and Hetal J. Vala 2013. A Review on Otsu Image Segmentation Algorithm.*International Journal of Advanced Research in Computer Engineering and Technology*2(2), ISSN 2278-1323

[11] P. K. Sahoo, S. Soltani, A. K. C. Wong, and Y. Chen 1988. A survey of thresholding techniques.*Computer Vision Graphics Image Processing*41,233-260.

[12]George J. Grevera 2007. Distance Transform Algorithms and Their Implementation and Evaluation.*Deformable Models, Biomedical and Clinical Applications, Springer* XVII

[13] Gunilla Borgefors 1986. Distance Transformation in Digital Images.*Comput. Vision, Graphics and Image Process*. 34, 344-371doi:10.1016/S0734-189X(86)80047-0

[14] G. Borgefors 1984. Distance Transformations in Arbitrary Dimensions.*Comput. Vision, Graphics Image Process*. 27, 321-345.doi:10.1016/0734-189X(84)90035-5

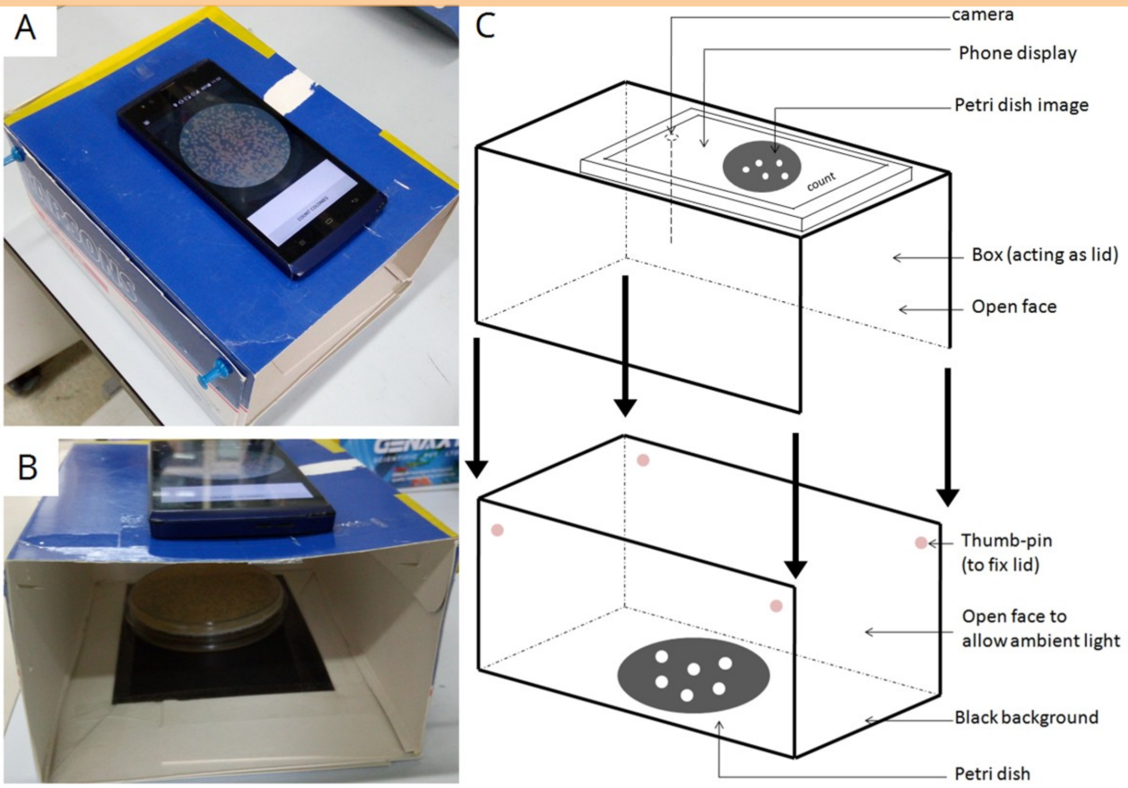

Fig 1. Alternative method to click petridish image outside the hood showing (A) top view of the cardboard box setup, (B) positioning the petri dish inside the box and (C) schematic illustrating the design method for the cardboard box

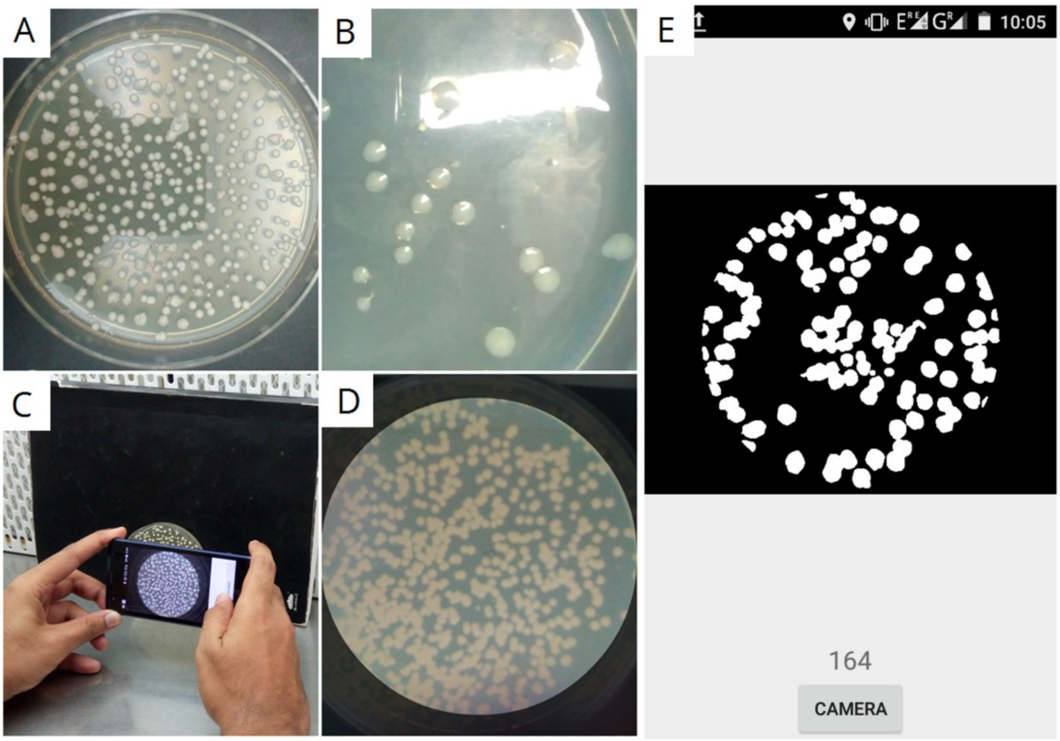

Fig 2. Imaging of colonies on a petri plate using a cellphone. (A) shadow of phone cast on petri dish while imaging, (B) unwanted reflections appearing from shiny edges of colonies and overhead light while taking image in ambient light conditions. (C) Suggested method to take picture inside the hood (D) cropping circle focused on the inside of the petri dish, and (E) Colonizer app user interface

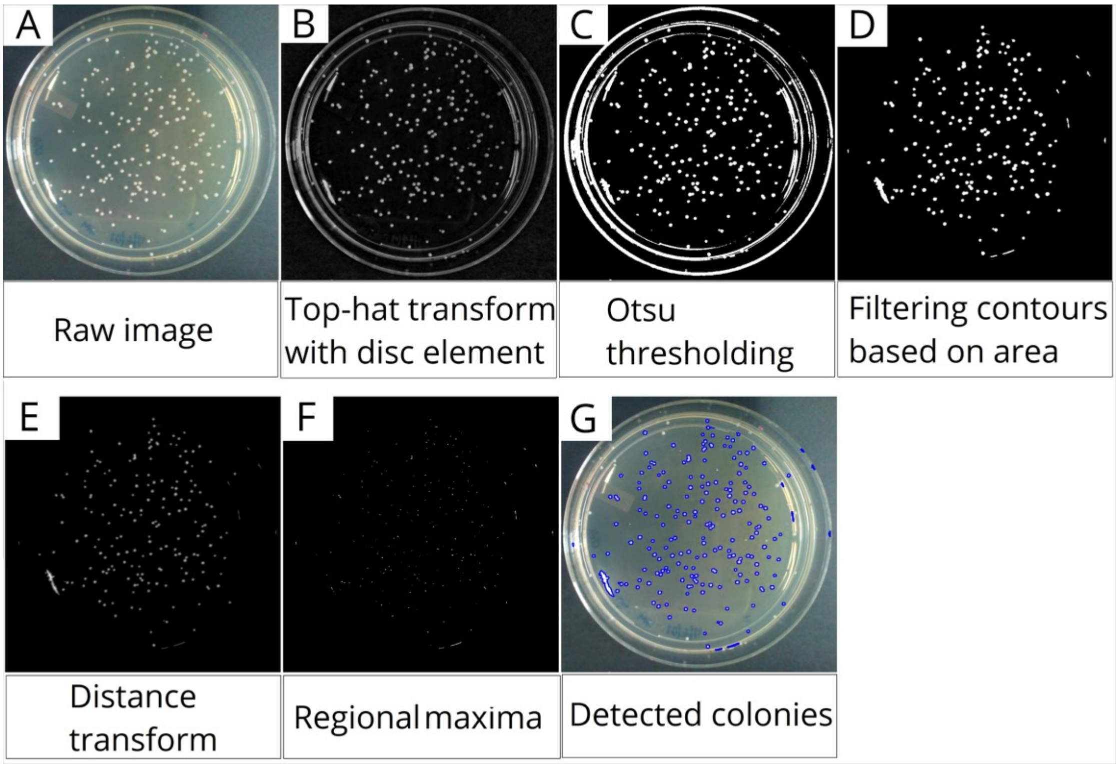

Fig 3. Step by step processing of the image by the Colonizer application showing the flowchart of the designed algorithm

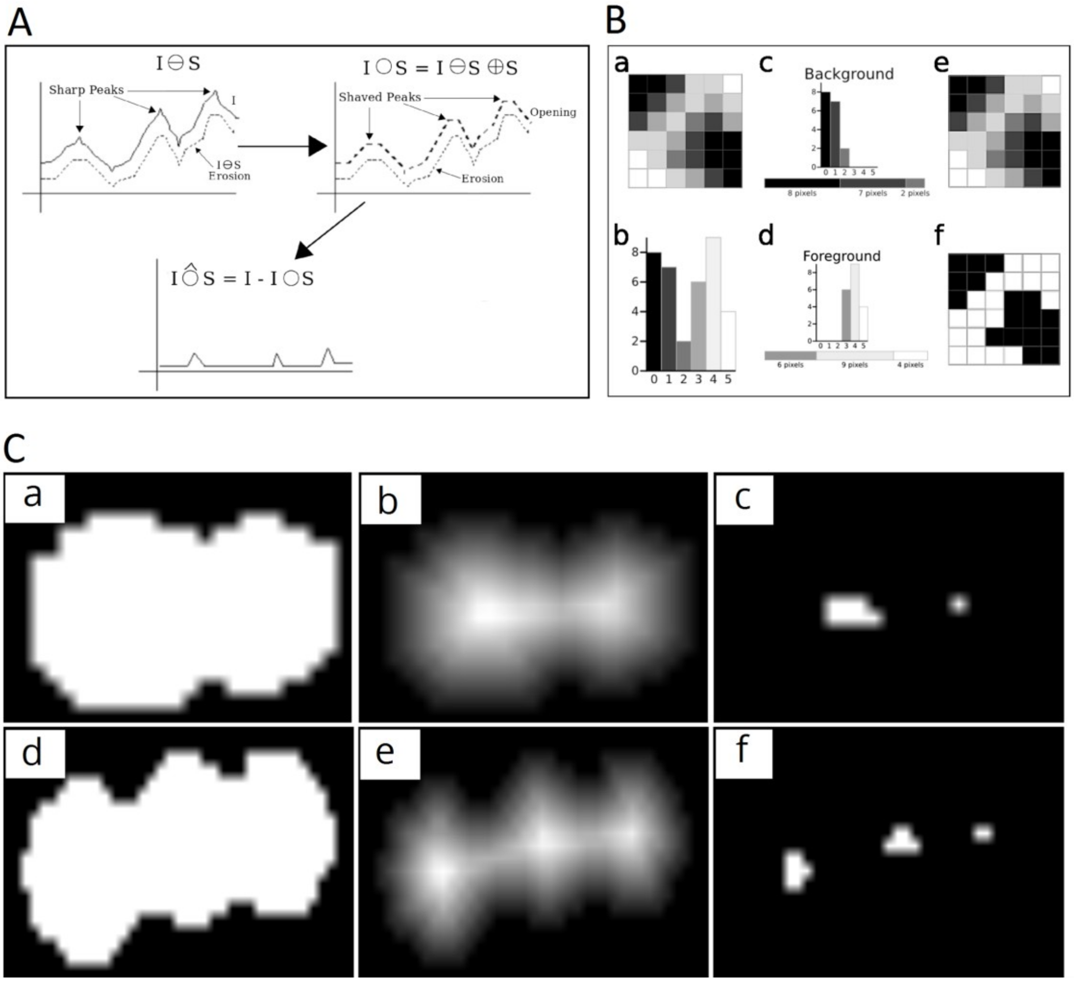

Fig 4. Illustration of the different features of the algorithm. (A) Top-hat filter: Image I is a chain of peaks with poor contrast between leftside peaks and rightside valleys. Erosion shaves of the peaks, dilation reconstructs the mountains without sharp peaks. Subtraction from the original image produces the peaks only. The poor contrast lighting condition thus has been eliminated using the tophat filter transform [adapted from Top-hat and Bottom-hat filters, *utam.gg.utah.edu*]. (B) Otsu's thresholding algorithm showing (a) 6-level grayscale image, (b) histogram of the original image, (c) background histogram, (d) foreground histogram, (e) – (f) original image thresholded by Otsu's algorithm to produce an image with sharp contrast between background and foreground [9]. (C) Implementation of a distance transform algorithm to segment connected colonies in an image. The figure shows (a) a double colony cluster, (b) distance transform of the original image, (c) finding regional maximum and hence segmenting out connected components and (d – f) similar process to segment out a triple colony cluster

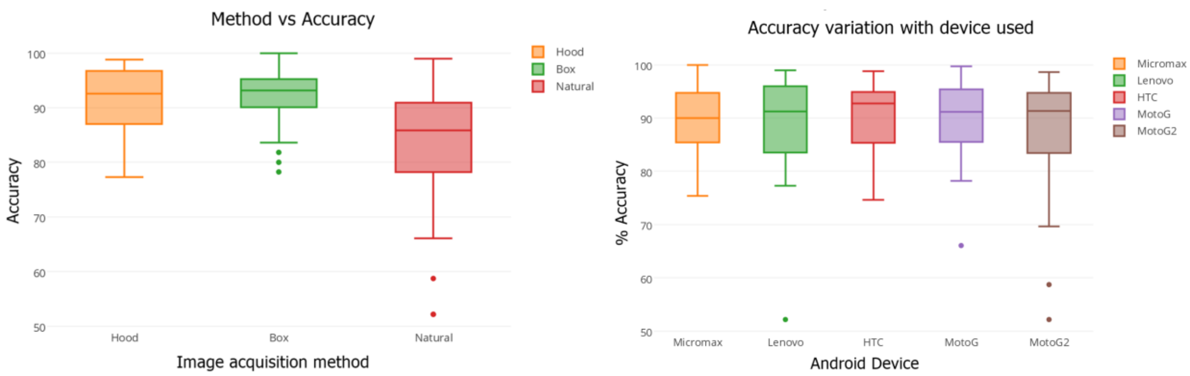

Fig 5 (A) Mean accuracy of the Colonizer app, using different lighting and imaging methods with the phone. Ambient lighting has lower median accuracy and larger variations than images taken in the box or the hood. (B) The median accuracy remains roughly the same and has low variations are seen across different phones, using a constant imaging method (cardboard box).

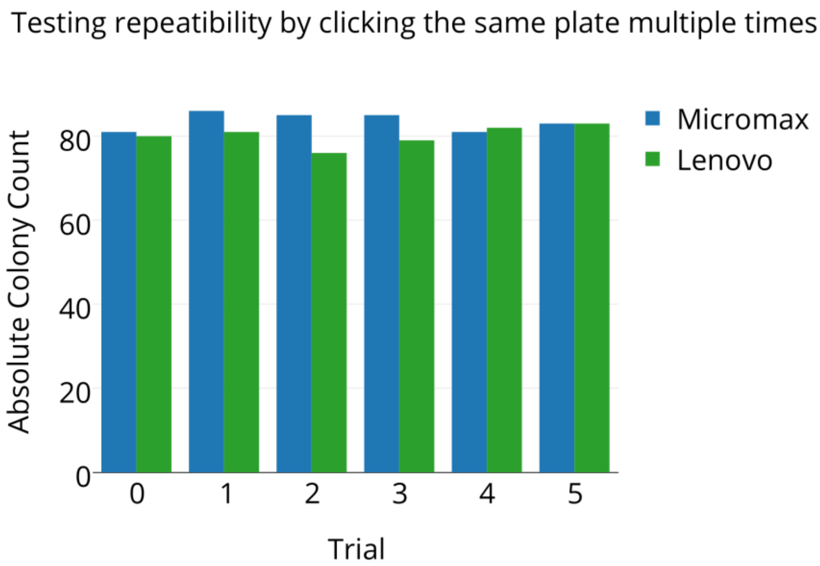

Fig 6. Repeatability experiment showing a single plate imaged and counted multiple times using a single phone. The experiment was performed in similar conditions for two different models of smartphones and shows a low variance of 4.7 and 6.1, respectively

Plate type	**Manual Count**	**Micromax**			**LENOVO**			**HTC**			**MOTO G**			**MOTO G 4 PLUS**		
		Hood	Box	Ambient	Hood	Box	Ambient	Hood	Box	Ambient	Hood	Box	Ambient	Hood	Box	Ambient
***S. cerevisiae* (yeast)**	63	66	63	77	53	61	61	57	60	59	55	61	67	56	60	89
	110	88	116	88	85	88	90	92	92	90	90	90	97	92	95	86
	221	188	191	189	192	197	175	216	209	165	207	203	146	204	209	154
	650	633	629	620	661	643	593	634	630	604	661	648	643	659	639	632
***E.coli* (bacteria)**	69	60	76	86	66	65	102	64	75	84	66	79	78	61	63	102
	170	160	186	187	165	158	188	168	183	194	178	207	185	173	159	199
	365	331	336	322	338	343	318	346	341	324	325	337	312	334	340	316

Table 1. Performance of the android app for colony counting on different colonies in multiple lighting conditions

PeerJ Preprints | https://doi.org/10.7287/peerj.preprints.2792v1 | CC BY 4.0 Open Access | rec: 8 Feb 2017, publ: 8 Feb 2017