JCB Journal of Cell Biology

TOOLS

Object detection networks and augmented reality for cellular detection in fluorescence microscopy

Dominic Waithe[1,2], Jill M. Brown[3], Katharina Reglinski[4,6,7], Isabel Diez-Sevilla[5], David Roberts[5], and Christian Eggeling[1,4,6,8]

Object detection networks are high-performance algorithms famously applied to the task of identifying and localizing objects in photography images. We demonstrate their application for the classification and localization of cells in fluorescence microscopy by benchmarking four leading object detection algorithms across multiple challenging 2D microscopy datasets. Furthermore we develop and demonstrate an algorithm that can localize and image cells in 3D, in close to real time, at the microscope using widely available and inexpensive hardware. Furthermore, we exploit the fast processing of these networks and develop a simple and effective augmented reality (AR) system for fluorescence microscopy systems using a display screen and back-projection onto the eyepiece. We show that it is possible to achieve very high classification accuracy using datasets with as few as 26 images present. Using our approach, it is possible for relatively nonskilled users to automate detection of cell classes with a variety of appearances and enable new avenues for automation of fluorescence microscopy acquisition pipelines.

Introduction

The microscopy image acquisition process can be highly repetitive and time consuming for the scientists who must be present throughout much of the process. Furthermore, experimental decisions made through this conventional acquisition process are difficult to describe and quantify, making the experiment hard to document and share scientifically. This lack of ability to communicate decisions means it is difficult for the scientific community as a whole to question and discuss methodologies and selection strategies, meaning we are at risk from the unconscious (and potentially conscious) bias of individuals. This issue is difficult to address but can be approached by adopting and introducing technology that allows improved documentation and reproducibility of data acquisition during an experiment.

There are a number of conventional high-content automated optical light microscopes that can find and image cells on the fly, relieving the effort of acquisition for the researchers and providing thorough documentation of the acquisition pipeline (Bellomo et al., 2017; Thomas, 2010). Techniques like this often use expensive hardware and computers and are often built on signal-processing methods for 2D or, more recently, 3D imaging platforms. Once created, these methods are powerful and fast, but they lack flexibility, and a skilled analyst is often required to tweak the parameters or modify the algorithms to detect a different cellular appearance, which can distance the user (e.g., the biologist) from the process of acquisition. An optimum solution represents hardware and algorithms that are easy to adapt by relatively unskilled users to recognize and localize cells of any type reproducibly and reliably and that can then be left, once validated, to perform bulk experimentation. Furthermore, for these approaches to be widely used, they must be deployable in an affordable and modular form and work in tandem with conventional microscopes and equipment.

Computer vision (CV) has developed to solve various challenges in video and photography (LeCun et al., 2015). In recent years, algorithms inspired from the CV domain have made a noticeable impact in the domain of microscopy image analysis, and interest continues to grow (Çiçek et al., 2016; Ronneberger et al., 2015; Schmidt et al., 2018 *Preprint*; Weigert et al., 2017). Object detection, a subdiscipline of CV, has developed with the goal of predicting bounding boxes for multiple objects in images or videos with potentially different classes and scales. In the

[1]Wolfson Imaging Centre Oxford, Weatherall Institute of Molecular Medicine, University of Oxford, Oxford, UK; [2]Medical Research Council Centre for Computational Biology, Weatherall Institute of Molecular Medicine, University of Oxford, Oxford, UK; [3]Medical Research Council Molecular Haematology Unit, Weatherall Institute of Molecular Medicine, University of Oxford, Oxford, UK; [4]Medical Research Council Human Immunology Unit, Weatherall Institute of Molecular Medicine, University of Oxford, Oxford, UK; [5]Nuffield Division of Clinical Laboratory Sciences, Radcliffe Department of Medicine, John Radcliffe Hospital, University of Oxford, Oxford, UK; [6]Institute of Applied Optics and Biophysics, Friedrich Schiller University Jena, Jena, Germany; [7]University Hospital Jena, Jena, Germany; [8]Leibniz Institute of Photonic Technology e.V., Jena, Germany.

Correspondence to Dominic Waithe: dominic.waithe@imm.ox.ac.uk.

https://doi.org/10.1083/jcb.201903166

past, this approach has been applied across many fields, including pedestrian detection, face detection, autonomous vehicles, and robotics (for reviews, see Andreopoulos and Tsotsos, 2013; Dollár et al., 2012; Li and Allinson, 2008; Li et al., 2015; Ruiz-del-Solar et al., 2018; Sun et al., 2006; Verschae and Ruiz-del-Solar, 2015). So far, object detection networks have not been used extensively for microscopy-based applications, though there are some recent contributions that use these type of networks (e.g., astrocyte detection; Schmidt et al., 2018 *Preprint*; Suleymanova et al., 2018). What potentially makes the object detection networks so attractive to microscopy is their accuracy, ease of use, and predictive speed. Thanks in part to the design of the more recent networks, these can be efficiently implemented on a graphics processing unit (GPU) and so can evaluate images in close to real time. This makes them perfect for use in an automated microscopy setup where a microscope will image and apply analysis in sequence. For these reasons, the possibility of using object detection networks in microscopy is an interesting one and worthy of investigation.

Faster-RCNN (region-based convolutional neural network) was the first network to combine features for region proposal with object classification and represents the culmination of a systematic set of advances and optimizations (Girshick, 2015 *Preprint*; Girshick, R., J. Donahue, T. Darrell, and J. Malik. 2014. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 580–587; Ren et al., 2015). Following Faster-RCNN, several other competitive algorithms have been developed that compete with and outperform Faster-RCNN in several aspects, including SSD (Liu et al., 2016), YOLO (Redmon, J., S. Divvala, R. Girshick, and A. Farhadi. 2016. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 779–788; Redmon and Farhadi, 2017 *Preprint*; Redmon and Farhadi, 2018 *Preprint*), and RetinaNet (Lin, T.-Y., P. Goyal, R. Girshick, K. He, and P. Dollár. 2017. Proceedings of the IEEE International Conference on Computer Vision. 2980–2988). Augmented reality (AR) is a technology that takes real-world scenes and enhances them through graphical annotation and works especially well when combined with CV techniques that work in real time. AR has been used for many years in jetfighter head-up displays and now is finding increasing usage in AR headsets (e.g., Microsoft HoloLens) and the automotive industry. Recently, AR has started to appear in microscopy (Chen et al., 2019; Edwards et al., 1999). In all its domains, AR brings the user in closer proximity to the information which can support them in their work and also reduces the distraction which can come with continually looking backward and forward from different independent displays or sources of information.

Because of the relative complexity of deep learning architectures and the high rates of data acquisition possible by modern sensors, there is currently a commercial trend toward developing cheap distributed hardware. An example application for this is a self-driving car, which acquires terabytes of data per day but must be able to analyze and respond to that information in real time. To respond to this need, companies like Nvidia have developed miniaturized and affordable GPU-enabled computers known as "Edge" devices. The Nvidia Jetson TX2 development board is one such example. The Jetson boards are compact, well defined, and affordable, meaning that they can be adopted and supported easily by an open community. These devices are used to distribute computation, allowing real-time processing within, for example, an autonomous car or, in our case, an autonomous microscope.

We show here that object detection networks are very suitable for fluorescence microscopy. We show that despite their complexity, these algorithms can be trained to work on relatively modest-sized training datasets. We characterize and contrast several object detection algorithms and determine which is best for application in microscopy. Further, we prototype and test an algorithm that can use the bounding boxes predictions from these networks to find and localize cells in close to real time in a 3D environment, a framework we call the autonomous microscope control algorithm (AMCA). We demonstrate AMCA working on a compact inexpensive Nvidia Jetson TX2 development board and show that the outputs of this analysis and acquisition can be visualized using an inexpensive display integrated with the microscope binoculars, making it much more appealing and immersive for the user. These techniques allow the user to automate their research in a highly customizable way but also allow them to question their assumptions and quantify and understand their sample earlier in the experimental pipeline.

Results

Choice of algorithm for cellular detection

An extensive collection of 10 datasets was established for this study. Each dataset was split into training and test sets, and each image was manually annotated with ground-truth regions (bounding boxes; example images are shown in Fig. 1, Fig. 3, Fig. 4, Fig. S1, and Fig. S4). With these data, we were able to thoroughly test and benchmark four deep learning object detection networks, specifically Faster-RCNN, YOLOv2, YOLOv3, and RetinaNet, and assess their performance on microscopy-acquired images. Many of the learned features in visual tasks are universal and can be applied with minimal tuning to different "objects" through a technique known as transfer learning (Hollandi et al., 2020; Pawlowski et al., 2016 *Preprint*; Stringer et al., 2020; Yosinski, J., J. Clune, Y. Bengio, and H. Lipson. 2014. Proceedings of Neural Information Processing Systems 27. 3320–3328). In our study, the object detection networks used had models that were pretrained on the photography image database ImageNet (Deng, J., W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. 2009. IEEE Conference on Computer Vision and Pattern Recognition. 248–255) and then fine-tuned for our application. Upon visual inspection, the predictions made by each of the networks on the test data were highly accurate with respect to the ground-truth regions created by manual annotation of the datasets (Fig. 1).

With six datasets, we evaluated the average precision (AP) of the algorithm at different iterations of training (Fig. S2) and under a number of different training regimens (TRs; Fig. S3). The four different TRs (TR1–TR4) represent different data-augmentation and training methodologies, AP and TR1–TR4 are described in detail in the Materials and methods section.

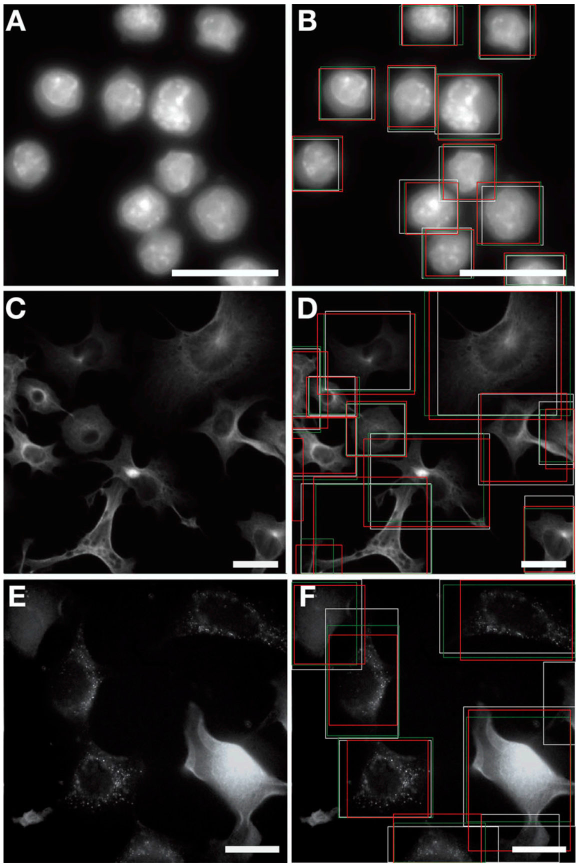

Figure 1. **Example fluorescence microscopy data generated for our study with corresponding ground-truth human annotations and object detection predictions.** All object detections >0.5 confidence are shown. **(A and B)** Eukaryotic cell dataset, fluorescently stained with DAPI. **(C and D)** Neuroblastoma cells fluorescently stained with GFP-phalloidin. **(E and F)** HEK cells fluorescently expressing GFP-SCP2 protein. Ground-truth boxes (white), YOLOv2 prediction boxes (red), and Faster-RCNN prediction boxes (green). Scale bars, 25 µm.

Deep learning networks typically have some kind of data-augmentation procedure to maximize the amount of data used for training. In addition to the normal methods, we additionally augmented the datasets by vertically flipping the microscopy images. This is possible with microscopy, as the images produced are rotationally invariant, as opposed to photographic images, where flipping vertically would yield an unnatural upside-down result. When we pooled the individually trained datasets from all the networks (Fig. 2 A, TR1 and TR2), we found a significant increase in accuracy when including additional vertically flipped training data (P = 0.016). Furthermore, we reasoned that training on jointly on several discrete (i.e., different cell types) but similar datasets would produce models that would generalize better and perform more accurately on individual cell classes (Hollandi et al., 2020). However, when the networks were jointly trained across different datasets and then evaluated on individual datasets (Fig. 2 A, TR1–TR3), we found no significant increase in accuracy (P = 0.172). It should be noted, however, despite no substantial boost in accuracy, it is still very attractive to train one model to recognize multiple classes rather than training a different model specifically for each class, as this saves a lot of computation time and memory.

To more easily compare the algorithms, we calculated their representative accuracy across each of the datasets by calculating the mean AP (mAP) of all the TRs (Table S1). Looking at the performance of each algorithm over the datasets (Fig. 2 B), we saw that overall YOLOv2 and RetinaNet were best performing, each performing best for three datasets. YOLOv3 performed significantly less well than the two best algorithms in all but two datasets. Compared with YOLOv2, v3 has been designed with a cluster of three output layers, each which make predictions at different spatial scales. Although this increased capacity to handle scale is beneficial in photography (Redmon and Farhadi, 2018 *Preprint*), this may not yield benefits in microscopy and therefore may be restrictive due to the added complexity. In summary, we saw that there was added accuracy benefits gained from using training sets augmented with additionally vertically flipped data, and we conclude that YOLOv2 was the preferred algorithm overall because of its speed advantage over RetinaNet (Redmon and Farhadi, 2017 *Preprint*), in addition to its high accuracy for our data.

Testing across different scales and optical resolutions

For collaborative research, and for using different microscopy modalities in concert, it may be necessary to train models on a set of data derived from one microscope and to make predictions based on data generated on a different system. We explored this idea and, specifically, the capacity of YOLOv2 to handle differences in digital and optical resolution. First, we explored different digital resolutions (Fig. 3, A–D). An image of COS-7 cells taken with 100× objective (Fig. 3 A) was resampled, using binning, to 50% (Fig. 3 B) and 20% (Fig. 3 C) of its initial resolution. Additionally, for three datasets (COS-7 nucleopore, C127 DAPI, and erythroid DAPI), we took models trained independently and then evaluated them on test data downsampled to lower resolutions (Fig. 3 D). Across the datasets, the AP was reduced by <1.5% at 50% resolution (0.5) and <15% at 20% resolution (0.2). We conclude that digital resolution had some impact on quality, especially at 20%, but this impact was not huge, given the dramatic reduction in pixels present. We reasoned that this resilience in YOLOv2 was because images are always preprocessed to fit the network input (416 × 416 pixels) and, as long as there was not too much loss of visual detail, then the accuracy of prediction was relatively well maintained. We also addressed how sensitive YOLOv2 was to the quality of objective (specifically the NA and magnification; Fig. 3, E–G). Here, COS-7 cells stained with nuclear pore were imaged on a Nikon microscope equipped with a 40× 0.55 NA and also a 10× 0.45 NA objective. The images were annotated with bounding boxes and evaluated with models (n = 3) previously trained on test images acquired using a 100× objective on an Olympus microscope. The object detection network performed well on the test dataset acquired with the 40× objective, yielding similar accuracy to the default 100× test dataset 0.978 ± 0.034 (AP ± SD). With the 10× objective, the images were first cropped to be the same physical scale as a 40× image (200 × 200 µm; Fig. 3 F, dashed red box). Upon assessment, the AP

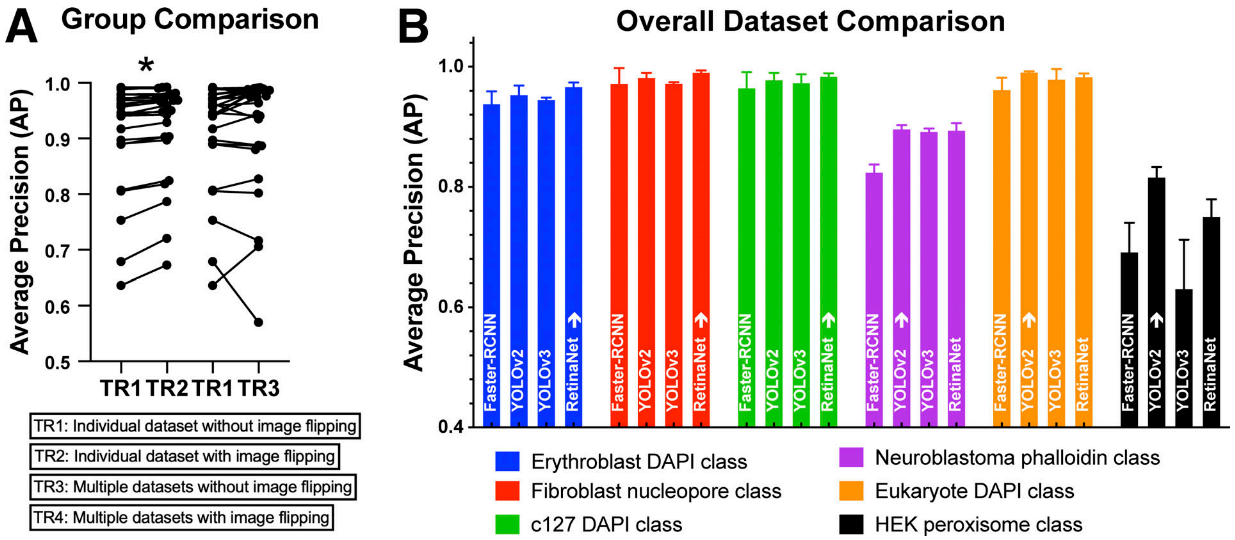

Figure 2. **Summary comparison of object detection algorithms for cellular detection. (A)** Comparison of AP for all datasets showing that additional vertical data flipping is effective for raising accuracy in general. Without (TR1) and with vertically flipped data augmentation (TR2) and when trained using multiple datasets without (TR3) and with (TR4) vertically flipped data. The Friedman's test was applied using Dunn's multiple comparisons to compare TR1/TR2 (*n* = 24, AP ± SD, * P < 0.05) and TR1/TR3 (*n* = 24, AP ± SD, nonsignificant). **(B)** Overall average AP comparison of Faster-RCNN, YOLOv2, YOLOv3, and RetinaNet for each dataset, averaging across each of the training modalities (TR1–TR4), described in A. Numerically, the best-performing algorithm is marked with an arrow for each dataset case YOLOv2 (3/6) and RetinaNet (3/6; *n* = 4, AP ± SD).

dropped by 85% (0.145 AP) even though these images represented a similar level of digital resolution with respect to the digital 0.2 scale (Fig. 3 D). We reasoned that this drop in AP was likely due to the lower NA and magnification of this objective and the resulting loss of fidelity of key details in those images that resulted. Interestingly, we found the accuracy of detection could be partially restored when the physical size of the cells in the image was close to that of the training dataset. When images were cropped to represent the same physical size (133.12 × 133.12 µm) as the training images (Fig. 3, E and F, dashed white boxes), the accuracy increased (Fig. 3 G). In conclusion, networks such as YOLOv2 will perform well across different microscopy platforms as long as the optical resolution is consistent and the physical dimensions of the images are matched.

Multiclass and multichannel data

We wanted to explore the network's ability to distinguish more than one class (or phenotype) of cell within a given image. To this end, we differentially labeled images with multiple cell classes in the human embryonic kidney (HEK) peroxisome dataset, where we compared cells with punctate staining only versus all cells present in the image ("all"; i.e., with punctate and diffuse/residual staining; Fig. 3, H and J). In addition, we used the erythroid DAPI all dataset (Fig. 3, I and J), where we trained for either single and multinucleate cells or for all cells (i.e., single/multinucleate and apoptotic cells. We found that the network was able to distinguish subtle phenotypes in both cases even in these challenging datasets, though with a lower accuracy (Fig. 3 J). The reduction in accuracy is a consequence of the subtlety of the task; it is easier to recognize all the cells present rather than to distinguish them based on subtle visual phenotypes. This capacity of multiclass detection is intrinsic to object detection networks and could be used to great effect in microscopy in the development of assays that search for rare phenotypes or dynamic imaging triggered in response to morphological change. We were also interested in how the object detection networks would perform on multichannel data (Fig. S4, A–C). Both the neuroblastoma phalloidin DAPI and erythroblast DAPI glycophorin A datasets were dual stained (Fig. S4, A and B). We found that YOLOv2 was clearly capable of recognizing two-channel images as compared with the single-DAPI-channel images and that this additional information was neither detrimental nor greatly beneficial to the classification performance in these cases (Fig. S4 C).

Integrated and automated image acquisition by object detection

Cells when viewed under a microscope are predominantly 3D, spanning more than one focal plane. A natural progression from localizing cells in 2D, therefore, is to localize them in a 3D environment. To adapt the object detection networks so that they could be used for acquiring cells in a 3D environment, we developed AMCA. AMCA is a Python-written control framework that interfaces with the microscope, camera, and control hardware to dynamically acquire images in 3D (Fig. 4 A). At its core is an object detection network (e.g., YOLOv2), which is used to inform the system whether there are cells present in a particular optical slice. Through custom Python scripts, it is possible to efficiently scan the slide (automated acquisition) and only acquire image volumes and slices where cells are identified (cellular detection). This is efficient, as only slices encompassing cells are retained and imaged and the microscope can quickly move through areas lacking cells, without continual prompting from the user.

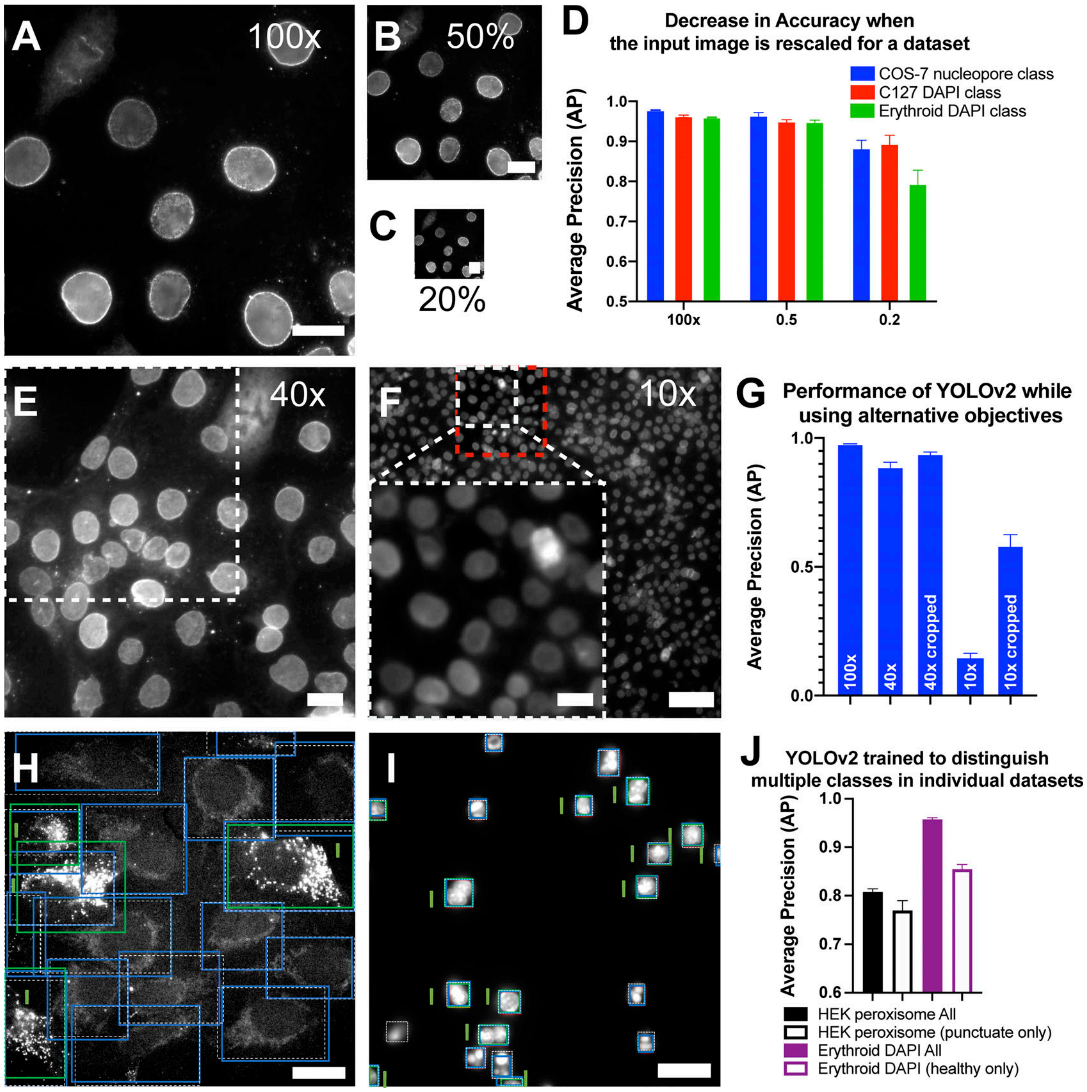

Figure 3. **YOLOv2 object detection performs well in a number of domains. (A–D)** YOLOv2 detection accuracy is consistent across images of lower digital resolution but is sensitive to optical resolution. **(A)** COS-7 cells stained for nuclear pore and imaged using a 1.4 NA 100× objective. **(B)** Same image as in A resampled at 50% of pixel resolution. **(C)** Same image as in A but resampled at 20% of pixel resolution. Scale (A–C) is 20 µm. **(D)** The accuracy of prediction on the 0.5 (B) and 0.2 scaled dataset (C) stays relatively high compared with normal resolution (1.0, 100×) when compared across three independent datasets (*n* = 3, mean ± SD). **(E)** COS-7 cells stained for nuclear pore and imaged with 40× 0.55 NA objective (scale bar, 20 µm). The dotted frame represents image area of equivalent physical scale of images A–C i.e., 133.12 × 133.12 µm. **(F)** COS-7 cells stained for nuclear pore and imaged with a 10× 0.45 NA objective (scale bar, 100 µm; inset, 20 µm). The white dotted inset is a zoom region on the dotted frame and represents an equivalent physical dimension of A (i.e., 133.12 × 133.12 µm); the red dotted frame represents same physical dimension as E (i.e., 200 × 200 µm). **(G)** Graph showing optical resolution is critical for performance of YOLOv2 when used to evaluate images collected on different microscopes with different objective types (40× and 10×; *n* = 3, mean ± SD). **(H–J)** Multiclass cellular detection in HEK peroxisome and erythroid DAPI datasets. YOLOv2 can be used to discretely identify cells with specific visual phenotypes within a single image. **(H)** HEK cells with varying levels of GFP-SCP2 expression, with either punctate fluorescence or a low level of diffuse nonpunctate fluorescence. **(I)** Erythroid nuclei stained with DAPI, highlighting either single/multinucleate cells that are healthy or in a state of apoptosis characterized by a blebbed appearance. In both images (H and I), white dashed ROIs represent ground-truth annotations used for training. ROIs with a star in close proximity represent the subset of annotations that were labeled positive for a phenotype of interest in the training. Blue ROIs represent predicted regions from model trained to recognize all cells in the image, whereas the green ROIs represent prediction for model trained to recognize a specific subset of cells present. **(H)** Output of an experiment with ROI predictions from a classifier trained to recognize all the cells in the image, while the second model (blue ROI) was trained to only recognize the cells exhibiting punctate fluorescence. **(I)** Output of an experiment where one network was trained to recognize only heathy single/multinucleate cells (green ROI), whereas the second model was trained to recognize all cells, including apoptotic cells (blue ROI). In perfectly classified images, green regions should only appear next to annotation regions with a green star (I has some wrongly classified regions). Scale bars represent 20 µm in both images. **(J)** Graph summarizes the AP measured with respect to the different conditions in H and I. Detector is capable of recognizing phenotypic subsets; however, the accuracy drops (in these experiments) when the network is trained to recognize a subset of cells (white bars) rather than cells in each image (filled bars; *n* = 3, mean ± SD).

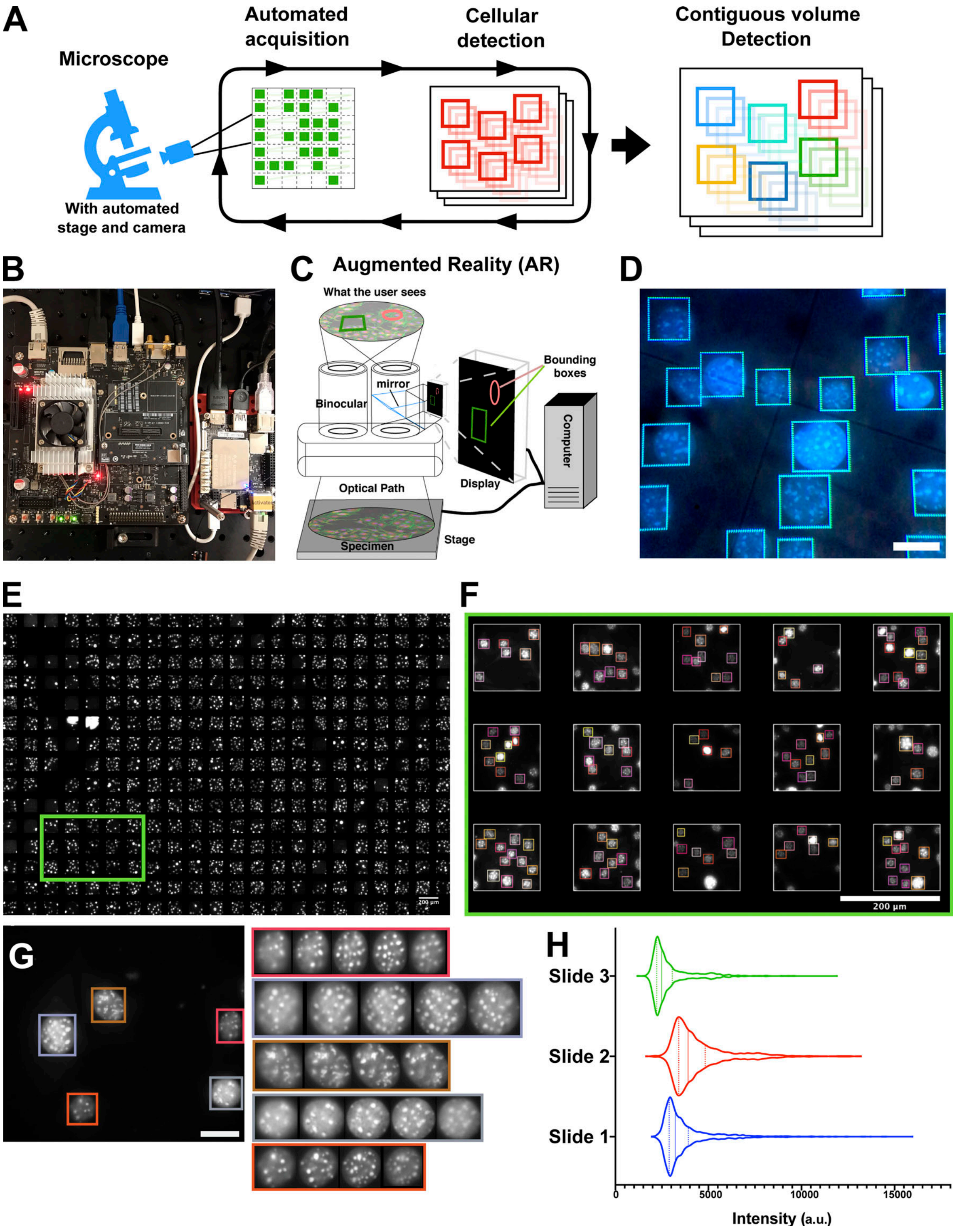

Figure 4. **Autonomous microscopy and AR powered using Jetson TX2 and LattePanda allows extensive and descriptive screens to be performed with ease. (A–D)** Schematic illustrating the AMCA working with a Jetson TX2 development board and AR optics. **(A)** The AMCA will control the automated acquisition on the microscope and will move the slide, imaging at different positions in sequence. If cells are detected in a location, the image will be stored and then the location optically sectioned in the "z" dimension until no more cells are detected. Once all the regions containing cells have been detected in a stage location, they are processed using a contiguous volume detection to find the 3D regions that encapsulate the individuals cells (see G). **(B)** Photo of the Jetson TX2 development board (left) on which the AMCA algorithm, object detection, hardware control, and image acquisition are running. On the right-hand side is the LattePanda Windows computer, which runs Windows-specific hardware control software and communicates with the Jetson TX2 via an ethernet cable. **(C)** AR allows the user to see the outputs of the analysis algorithm when viewing the sample down the microscope. The ROI generated from cellular detection can be visualized through the AR system as the microscope acquires images online or subsequently offline, when the user views areas of the sample that have already been processed. **(D)** View down the binocular eyepiece of the microscope where the AR graphics are overlaid with the light emitted from the sample (scale bar, ~20 µm). **(E–H)** Preliminary screen of C127 cells stained with DAPI. **(E)** Low-resolution overview image of C127 cells acquired during screen. Scale

bar, 200 µm. **(F)** Zoom area shown by green rectangle in E. Image areas are shown with detected cells bounded by colored boxes. Cells touching the image area boundaries are excluded in this analysis. Scale bar, 200 µm. **(G)** Left: Example image with bounding boxes representing discrete cellular classification from object detection algorithm and the color represents track linking with the contiguous volume detection (colored rectangles). Depiction of cell classifications tracked through the different z-slices; color border represents cells in G. **(H)** Summary violin plots calculated over the mean intensity values for the cells acquired during the screen. Three independent slides were screened and analyzed (green, red, and blue). The median value (3,216, 3,905, and 2,487, solid line), the lower quartile (2,891, 3,404, and 2,227, left dashed line) and the upper quartile (3,923, 4,825, and 3,055, right dashed line) for each slide (1–3), respectively.

To be as affordable and accessible as possible, we used a Nvidia Jetson development board as the main control computer of the system. This system has all the desired functionality needed to control the system and is much more affordable than a full PC; configuring the system is also straightforward. It has a powerful GPU and also comes with a central processing unit and random access memory (RAM) and all the necessary functionality for a computer. The Jetson could handle most of the functionality and interactivity required; however, some Windows functionality was required to run some of the drivers for the specialist hardware. We therefore implemented an inexpensive LattePanda Windows development board. This is a much weaker computer than the Jetson but comes preinstalled with Windows 10. We used the LattePanda in concert with the Jetson to control the microscope acquisition (Fig. 4 B). By using this standardized inexpensive system, we hope to attract a large user base for this system and can support them directly and effectively as users can easily setup the same base hardware.

Here, we show that AMCA can dynamically acquire images in 3D in an automated fashion. Importantly, we demonstrate that the object detection network is fully integrated into the data acquisition workflow and runs in close to real time alongside it. Using our Jetson TX2 Development board and an input resolution of 416 × 416 pixels for the YOLOv2 object detection network, we achieved 4–5–frames per second output speed (~150–200 ms for the detection/control and allowing 50 ms for exposure). Detections are therefore instantly visible with this system and can greatly assist in the process of acquisition of the desired cell states/phenotypes.

AR provides visual feedback in real time

The experience of using AR in the context of microscopy enriches the experience and helps integrate the acquisition phase with sample analysis. Using off-the-shelf components and optical elements, we used a simple solution to provide AR within the microscope binoculars (see Materials and methods for more details; Fig. 4 C). The AR system lends itself very well to work alongside AMCA, which provided near-real-time analysis and processing of the object detection network. The AR system can work in two different ways: (1) "online" mode (Fig. 4 D and Videos 1 and 2), where the system simultaneously analyzes as you manually move around the slide; or (2) "offline" mode, where the slide has already been imaged and analyzed and you review the detections in the context of the images of the sample.

We wanted to showcase the potential for AMCA to perform screens on volumetric acquisitions of cells. For this, we used C127 cells, where nuclei were stained with DAPI after having been treated with a technique known as RASER-FISH (Brown et al., 2018). Using a 100× objective, we screened 624 imaging positions, arranged with uniform spacing of 200 µm between positions, across three slides. A low-resolution overview image of one of the slides is shown in Fig. 4 E, and a higher-resolution image is shown in Fig. 4 F. Uniquely localized cells were classified with colored bounding boxes (Fig. 4 G). The classification persists through the in-focus volumetric region, and each classification is linked with its counterparts from the same cell, denoted by the same color (Fig. 4 G, right). This clearly shows that object detection algorithms, when applied to microscopy, offer significant possibilities with regard to the identification and extraction of cellular subvolumes. As an overview, image volumes were "maximum" projected and the average intensity measured in each cell area (Fig. 4 H; 2,105, 2,100, and 1,887 cells for the first, second, and third slides, respectively), and this took ~34 min per slide. The resulting distributions have one main peak and an extended tail that suggests that there is more than one component contributing to the overall intensity distribution of the DAPI-stained cells. What this preliminary experiment does show is the potential power of this higher-throughput automated volumetric imaging approach to reveal data not usually visible from assays run across a small number of cells. By using AMCA, we were also able to optimize the acquisition owing to the dynamic analysis allowed by the objection detection algorithm. This dynamic analysis system cut short imaging of any volumes where the sectioning had already extended across all the cells present, saving valuable time. For large screens, this represents a considerable time-saving factor and also requires less storage.

Discussion

In this study, we created a publicly available collection of datasets of cellular images (http://doi.org/10.5281/zenodo.3894389) and annotated the identified cells therein with bounding boxes. This provided an excellent resource for training, benchmarking, and improving the object detection algorithms in this study and hopefully future studies.

We have comprehensively benchmarked four popular object detection algorithms (Faster-RCNN, YOLOv2, YOLOv3, and RetinaNet) for the task of cellular classification and localization. We found YOLOv2 to be an excellent choice for accurate cell detection, even with challenging data, and it also is the fastest. Furthermore, we found that we could enhance the performance of these algorithms with additional data augmentation in the form of vertical flipping. The future for these kind of networks within the imaging sciences is likely to revolve around generating network designs that can outperform YOLOv2 in terms of accuracy and flexibility. The main bottleneck for the acquisition process is no longer the speed of processing but the exposure time of the camera. There is certainly convenience to screening a sample and then performing fine-grain analysis with the same objective, but there are limits to the throughput achievable with high-magnification optics due to the limited field of view. A superior system would automatically correlate measurements made using a low-magnification "screening" objective with those of a higher-magnification immersion objective. Such a system would allow the user to accurately and quickly perform high-resolution imaging on a specific cell (or region) with a complete understanding of how that cells fits into the overall distribution of the cells in the sample.

In this work, we have developed a dynamic 3D acquisition control framework, AMCA, that interfaces with an automated stage and fast-acquisition camera to acquire image volumes. The object detection algorithm at the heart of AMCA allows near–real-time identification of cells and their efficient acquisition. Importantly, we demonstrated that multiple phenotypic classes of cells could be classified within images showing that AMCA can be extremely helpful in scanning for rare or particular cell types. This powerful technique can easily be trained by nonskilled users due to its simplicity. To aid in the adoption of AMCA and this technique in general, we have developed the system to work on an inexpensive Nvidia Jetson TX2 and LattePanda Development boards. This makes the system very affordable, easy to install, and easily applied to other systems as the installation steps will work for all Nvidia and equivalent boards. The development boards are also physically small and could potentially be incorporated directly into the microscope housing, making it possible to have an intelligent acquisition system embedded in the microscope, reducing the overall footprint of a system. We believe that such ease of accessibility of the system will make it a popular choice among scientists developing their own automated solutions. Using bounding boxes as a form of annotation by AMCA has the advantage that it is simple and quick to create and interpret. One criticism of using bounding boxes, however, is that cellular image analysis generally involves some form segmentation to discretize each cell and potentially reduce the impact of background pixels. Using bounding boxes, however, does not exclude this type of analysis and actually is an excellent prior for applying subsequent analysis methods through providing demarcation of the image.

Computer-based exploration of 3D data can often be cumbersome and unintuitive. In contrast, a microscope is a well-designed tool for navigating a 3D space and lends itself well to 3D exploration. By using AR in the visual output of the microscope, we have enabled the data generated by the microscopy system to be displayed within the context of the physical specimen. This is an enjoyable and intuitive experience for the user and allows for the quick comparison of areas that have been imaged, allowing one to understand it better. AR is a very attractive feature of our system, especially as it has been developed using conventional optical components. We envisage this type of technology becoming commonplace in microscopy and will work toward more compact and convenient implementations.

Materials and methods

Dataset generation

Our goal for applying object detection networks to microscopy was ultimately so that these algorithms could be applied to find and isolate cells within a 3D environment. As they stand, object detection algorithms are predominantly used to find objects in single 2D photography images or movies, and the training material is supplied to the algorithm exclusively in a 2D format (Andreopoulos and Tsotsos, 2013; Dollár et al., 2012; Li and Allinson, 2008; Li et al., 2015; Ruiz-del-Solar et al., 2018 *Preprint*; Sun et al., 2006; Verschae and Ruiz-del-Solar, 2015). Single-plane images are far easier to label by users than 3D volumes, requiring only a 2D bounding box to be placed around examples within the image. Therefore, we wanted to establish our methodology for microscopy, including training and prediction, in 2D and then apply it in a 3D environment. To validate the object detection algorithms, we created six different cell-based datasets and modified the networks so that they could be trained on these data and also validated against holdout test data (i.e., not used for training). Each dataset was divided into train and test datasets, and the object detection networks were trained and evaluated on the train and test datasets, respectively. With the exception of the neuroblastoma phalloidin data, each dataset was created and imaged within our host institution using conventional wide-field microscopes. The neuroblastoma phalloidin data were generated from an online resource (Yu et al., 2010) and the ground-truth segmentations converted into bounding box representations. These data in their entirety, as well as the annotations, are available in the repository (http://doi.org/10.5281/zenodo.3894389). The collection includes the following data.

Erythroblast DAPI (+glycophorin A)

Erythroblast cells were stained with DAPI and glycophorin A protein (CD235a antibody, JC159 clone; Dako) and Alexa Fluor 488 secondary antibody (Invitrogen). DAPI staining was performed using VectaShield Hard Set mounting solution with DAPI (Vector Lab). The number of images used for training was 80, and the number used for testing was 80. The average number of cells per image was 4.5.

Neuroblastoma phalloidin (+DAPI)

Images of neuroblastoma cells (N1E115) stained with phalloidin and DAPI were acquired from the Cell Image Library (Yu et al., 2010). Cell images in the original dataset were acquired with a larger field of view than our system, and so we divided each image into four subimages and also created region of interest (ROI) bounding boxes for each of the cells in the image. The images were stained for FITC-phalloidin and DAPI. The number of images used for training was 180, and the number used for testing was 180. The average number of cells per image was 11.7.

Fibroblast nucleopore

Fibroblast (GM5756T) cells were stained for a nucleopore protein (anti-Nup153 mouse antibody; Abcam) and detected with anti-mouse Alexa Fluor 488. The number of images for training was 26, and the number used for testing was 20. The average number of cells per image was 4.8.

Eukaryote DAPI

Eukaryote cells were stained with DAPI and fixed and mounted in Vectashield (Vector Lab). The number of images for training was 40, and the number used for testing was 40. The average number of cells per image was 8.9.

C127 DAPI

C127 cells were initially treated with a technique called RASER-FISH (Brown et al., 2018), stained with DAPI, and fixed and mounted in Vectashield (Vector Lab). The number of images for training was 30, and the number used for testing was 30. The average number of cells per image was 7.1.

HEK peroxisome all

HEK-293 cells expressing peroxisome-localized GFP-SCP2 protein were transfected with a GFP-SCP2 encoding plasmid, which contains a PTS-1 localization signal that redirects the fluorescently tagged protein to actively importing peroxisomes (Stanley et al., 2006). Cells were fixed and mounted. The number of images for training was 55, and the number of images for testing was 55. Additionally, we subcategorized the cells as "punctated" and "nonpunctated," where punctate represented cells that have staining where the peroxisomes are discretely visible and nonpunctated represented diffuse staining within the cell. The HEK peroxisome all dataset contains ROI for all the cells. The average number of cells per image was 7.9. The HEK peroxisome dataset contains only those cells with punctate fluorescence. The average number of punctate cells per image was 3.9.

Erythroid DAPI all

Murine embryoid body–derived erythroid cells, differentiated from murine embryonic stem cells, were stained with DAPI and fixed and mounted in Vectashield (Vector Lab). The number of images for training was 51, and the number of images for testing was 50. Multinucleate cells are seen with this differentiation procedure. There is a variation in size of the nuclei (nuclei become smaller as differentiation proceeds). The smaller, "late erythroid" nuclei contain heavily condensed DNA and often have visible heavy "blobs" of heterochromatin. Apoptotic cells are also present, with apoptotic bodies clearly present. The erythroid DAPI all dataset contains ROI for all the cells in the image. The average number of cells per image was 21.5. The subset erythroid DAPI contains non-apoptotic cells only. The average number of cells per image was 11.9.

COS-7 nucleopore

Slides were acquired from GATTAquant. GATTA-Cells 1C are single-color COS-7 cells stained for nuclear pore complexes (Anti-Nup) and Alexa Fluor 555 Fab(ab′)2 secondary stain. GATTA-Cells are embedded in ProLong Diamond. The number of images for training was 50, and the number for testing was 50. The average number of cells per image was 13.2.

COS-7 nucleopore 40×

The same GATTA-Cells 1C slides (GATTAquant) as above were used, but they were imaged on a Nikon microscope with a 40× NA 0.55 objective. The number of images for testing was 11. The average number of cells per image was 31.6.

COS-7 nucleopore 10×

The same GATTA-Cells 1C slides (GATTAquant) as above were used, but they were imaged on a Nikon microscope with 10× NA 0.45 objective. The number of images for testing was 20. The average number of cells per image was 24.6 (the entire field of view was not used).

Dataset annotation

Datasets were annotated by a skilled user. These annotations represent the ground-truth of each image with bounding boxes (regions) drawn around each cell present within the staining. Annotations were produced using Fiji/ImageJ (Schindelin et al., 2012) ROI Manager and the OMERO (Allan et al., 2012) ROI drawing interface (https://www.openmicroscopy.org/omero/). The dataset labels were then converted into a format compatible with Faster-RCNN (Pascal), YOLOv2, YOLOv3, and also RetinaNet. The scripts used to perform this conversion are documented in the repository (https://github.com/dwaithe/amca/tree/master/scripts/).

Microscopy setup

The fibroblast nucleopore, eukaryote DAPI, C127 DAPI, HEK peroxisome, erythroid DAPI, and COS-7 nucleopore cell datasets were acquired on an Olympus IX73 microscope with a 100× UPlanSApo NA 1.4 objective. The microscope was also equipped with a Photometrics Prime sCMOS camera (6.5 × 6.5 µm pixel, 2,048 × 2,048 chip), a CoolLED Ultra pe300 LED light source, an Applied Scientific Instrumentation automated xy stage, and a Physik Instrumente Piezo (P-733 2CL). Typically, 2× binning was applied to the camera (13 × 13 µm) and later and additional 2 × 2 digital binning. The erythroblast DAPI dataset was acquired on a DeltaVision Elite (GE Healthcare Life Sciences) equipped with an Olympus 60× NA 1.42 lens, filters for DAPI (excitation 390 nm, emission 435 nm) and FITC (excitation 475 nm, emission 525 nm) and a CoolSNAP HQ2 camera. The neuroblastoma phalloidin +DAPI cell line was acquired on a Zeiss Aviovert 200 microscope with filters for DAPI and FITC (Yu et al., 2010). The COS-7 nucleopore 40× and 10× datasets were acquired on a Nikon Eclipse TE300 microscope. The Nikon was equipped with a Excelitas XCite 120Q light source, a QImaging Rolera EM-C2 (electron multiplying charge-coupled device) camera, 8 × 8 µm pixel size, 1,004 × 1,002 chip, and a 40× long working distance phase 1 NA 0.55 and a 10× PlanApo differential interference contrast NA 0.45 objective. All acquisition experiments were performed at room temperature and pressure.

AR modifications

To develop the AR setup, we adapted commercial components with custom parts. The AR effect is created through the merging of the image emanating from a display screen with the image emanating from the microscopy sample. This was achieved through the coupling of a 50:50 beam splitter (BSW10R; Thorlabs) into the light path of the microscope. This was realized through adapting a Mightex Dichroic/filter cube (DSI-CUBE-OL-UA) to fit in between the observation tube and the observation tube mount of an IX73 microscope (Olympus). The Mightex Dichroic filter has the required circular dovetail mounts to fit within the binocular of the IX73 system, but in its default configuration, the beam splitter couples light toward the specimen and not the observer, which is what we require for the AR system. To correct this, we engineered two adapter plates to reverse the gender of the mounts, and details of these plates can be found in Data S1. The computer screen (HDMI 8” IPS LCD Screen Kit; Pimoroni) was positioned to the right of the microscope so that the base of the screen was parallel to the Mightex cube and perpendicular to the light path through the microscope. The screen was secured at the desired angle using standard M6 Post components (Thorlabs) and an Ailun Tripod Mount Adapter (B071XHYG5R; Amazon). Attached to the Mightex cube, between the beam splitter and light coming from the computer screen, was a 300-mm biconvex lens (LB1779; Thorlabs), which converged the light from the screen onto the beam splitter. The computer screen was placed ∼30 cm from the beam splitter, which resulted in an in-focus view of the screen graphics when looking down the binocular. An optional modification we made to the conventional IX73 setup to use a 50:50 beam-splitter cube in place of the mirror that directs light either to the camera or to the binocular. We made this modification so we could simultaneously view the specimen down the binocular and also record the same image on the computer. For this, we engineered our own cube holder (Data S1) and using superglue adhesive attached a 30-mm 50R/50T Standard Cube beam splitter (#32-701; Edmund Optics).

Benchmarking computer hardware

Benchmark experiments were run on Dell PowerEdge R730 Server (2x Intel Xeon E5-2650, 256 RAM, Nvidia Tesla K80 GPU) as well as on a Dell Precision Tower, with 32 GB RAM, Nvidia Quadro P5000 16 GB GPU, Dual Xeon Processor E5-2637, both with CentOS 7 installed.

Object detection algorithms

In this study, we took four leading publicly available object detection networks (Faster-RCNN, YOLOv2, YOLOv3, and RetinaNet). We found that typically, a peak level of accuracy was reached before the accuracy stabilized to a consistent value, and so all comparisons were made at the optimal number of iterations of training for each algorithm.

The code used for the Faster-RCNN is a tensorflow implementation and was modified from dBeker/Faster-RCNN-TensorFlow-Python3.5; it can be found at https://github.com/dwaithe/Faster-RCNN-TensorFlow-Python3.5. Faster-RCNN was configured as follows. The VGG16 network was used to initialize the classification layers. The parameters for learning were configured as follows: ‘Weight_decay’ = 0.0005, ‘learning_rate’ = 0.001, ‘momentum’ = 0.8, ‘gamma’ = 0.1, ‘batch_size’ = 256, ‘max_iters’ = 40,000, ‘step_size’ = 30,000. The network was modified to flip images not only horizontally but also vertically during data augmentation.

YOLOv2 was cloned from the source (https://github.com/AlexeyAB/darknet) and modified for this work (https://github.com/dwaithe/darknet3AB). The modified YOLOv2 network was run with configuration settings (yolov2_dk3AB-classes-#-##flip.cfg): ‘batch’ = 64, ‘subdivisions’ = 8, ‘height’ = 416, ‘width’ = 416, ‘channels’ = 3, ‘momentum’ = 0.9, ‘decay’ = 0.0005, ‘angle’ = 0, ‘saturation’ = 1.5, ‘exposure’ = 1.5, ‘hue’ = 0.1, ‘learning_rate’ = 0.001, ‘burn_in’ = 1000, ‘max_batches’ = 10000, ‘policy’ = steps, ‘steps’ = 4500, 4800, ‘scales’ = 0.1, 0.1. The network was modified to flip images not only horizontally but also vertically during data augmentation.

YOLOv3 was also cloned from the source (https://github.com/AlexeyAB/darknet) and modified for this work (https://github.com/dwaithe/darknet3AB). It was run with the following configuration settings (yolov3_dk3AB-classes-#-##flip.cfg): ‘batch’ = 64, ‘subdivisions’ = 16, ‘width’ = 608, ‘height’ = 608, ‘channels’ = 3, ‘momentum’ = 0.9, ‘decay’ = 0.0005, ‘angle’ = 0, ‘saturation’ = 1.5, ‘exposure’ = 1.5, ‘hue’ = .1, ‘learning_rate’ = 0.001, ‘burn_in’ = 1,000, ‘max_batches’ = 10,000, ‘policy’ = steps, ‘steps’ = 9,000, 9,600, ‘scales’ = 0.1, 0.1. The network was modified to flip images not just horizontally but vertically during data augmentation. The number of classes was set to 1 or to 6 and the filters adjusted accordingly to 18 or 33 respectively. (filters=(classes + 5)*3). YOLOv3 has three output layers, representing different scales, and the number of filters was corrected in each case.

RetinaNet was cloned from the source (https://github.com/fizyr/keras-retinanet) and was modified for this work (https://github.com/dwaithe/keras-retinanet) and run with settings: ‘batch’ = 1, ‘lr’ = 0.00001, ‘epochs’ = 50, ‘steps’ = 10,000, ‘backend’ = resnet50, ‘image-min-side’ = 800, ‘image-max-side’ = 1,333. The network was modified to flip images not only horizontally but also vertically during data augmentation. The number of classes was defined through the ‘retina_classes.csv’ file, which accompanies the training data.

Evaluation metrics

AP is a commonly used metric for assessing the accuracy of algorithms that are performing classification and/or localization. For this study, we use the updated VOC2010 AP definition described previously (Everingham et al., 2015) and as follows. In a given 2D image, containing one or more objects (i.e., cells), a trained object detection network will predict bounding regions for each of the objects contained within the image and associate a level of confidence (0–1.0) with that prediction. At a low-confidence threshold, many regions will be predicted whereas a higher confidence far fewer will, normally for visualization of results we show the predictions with a specific cutoff for each algorithm. For comparison between algorithms, however, we need to evaluate performance across confidence levels. The first stage in this process is to assess which of the predicted regions (B_p) is overlapping the ground-truth regions (B_{gt}). Those detections that have an overlap coefficient (a_0) of >50% are considered correct detections (true positives [TPs]); otherwise, they are defined as false positives (FPs):

$$a_0 = \frac{area\left(B_p \cap B_{gt}\right)}{area\left(B_p \cup B_{gt}\right)},$$

where $B_p \cap B_{gt}$ is the intersect and $B_p \cup B_{gt}$ is the union of these regions. Multiple detections are then ordered in terms of decreasing confidence. Multiple positive detections of the same region will only count the first detection as a positive and the rest as negative detections (false negatives [FNs]). If a ground-truth region contains no detections, this counts as a FN also.

There are no true negative values, as background regions are never actively identified. For a given class, a precision–recall curve is computed from a method ranked based on confidence. The precision is then calculated (precision = TP/(TP + FP)), and the recall (recall = TP/(TP + FN)) across all the data at each rank. We simplify the data by taking the maximum precision for any recall value, which results in the generation of a precision/recall curve, the area under which we can use to compare different algorithms. The so-called AP metric is achieved by taking the maximum precision across all recall values and taking the average. The metric mAP represents the mean AP value yielded from evaluation over different classes (i.e., performance across different cell types or objects).

Four different TRs (TR1–TR4) were formulated for the validation and comparison of each of the object detection networks: (1) In the naive approach, TR1, the training component of a single dataset was taken and used to train the network. The normal data augmentation for that network was applied with no additional step. (2) In the second approach, TR2, again the training component of a single dataset was taken and used to train the network, but this time, additional data augmentation was performed on the training data. Training images were also vertically flipped during the data augmentation, creating a larger set of training images. (3) In the third approach, TR3, we trained not just one dataset but multiple datasets simultaneously. A single model produced by this method could be then be used to evaluate images from each of the testing datasets. For the third approach, no additional data augmentation was applied. (4) Finally, in the fourth approach, TR4, we combined the TRs of the second and third approach. The networks are trained on multiple datasets simultaneously, and the data are also additionally vertically flipped during the data augmentation.

Statistical tests

Within the architecture of Faster-RCNN, YOLOv2/3, and RetinaNet, there are points at which randomness is injected into the training process. For example, the way the bounding boxes are selected and how they are shuffled for the training are done randomly. Much of this is unseeded in that it is nonreproducible. This will mean that every time a network is trained, the resulting model will be slightly different, and the model will see different training data at different points of training. As a consequence, the accuracy of trained models will vary slightly when retrained. If the variance is very high between these models, this suggests that the training procedure and model are not well optimized for the task. In this situation with relatively small amounts of data, some variance was likely. To gain awareness of the network stability, each experiment was repeated three times, the AP was measured and averaged, and the SD was calculated.

Statistics were calculated using GraphPad Prism software (v8.3.1). Normality tests (Shapiro–Wilk) were applied to the data in Fig. 2 A and Fig. S3, E–H, and the data were found to be not normally distributed. Nonparametric Friedman's tests were therefore used with Dunn's multiple comparison tests applied to the data in these cases.

Innovative acquisition control hardware

The computer used to control the microscope and manage the acquisition is a relatively inexpensive Nvidia Jetson TX2 development board. Along with a powerful central processing unit and RAM configuration, this system is equipped with a powerful GPU essential for running the deep learning object detection networks at speed. The Jetson has Ubuntu OS installed and can be connected to a monitor and keyboard/mouse like a regular computer. The Jetson TX2 is connected to the Photometrics Prime scientific complementary metal oxide semiconductor camera by universal serial bus 3 (USB3) interface. The CoolLED light source was connected via transistor–transistor logic cables to the Photometrics camera, which allows for fast triggering of the light source via the camera, however the lamp could also be controlled directly from the Jetson via USB cable. The Applied Scientific Instrumentation automated xy stage was connected to the Jetson via a standard USB–serial interface. For the Physik Instrumente Piezo, because the drivers written to control this software are available only for the 64-bit Windows operating system, we used an inexpensive Windows 10–installed LattePanda Development Board (https://www.lattepanda.com/) and connected this to the Jetson via an ethernet cable. This allowed us to use the specialist drivers of the Piezo in their native environment while taking advantage of the GPU power of the Jetson for all other functions. Full details of this system and how it was connected can be found at https://github.com/dwaithe/amca/tree/master/jetson.

3D acquisition algorithm

AMCA is written in Python and is freely available (https://github.com/dwaithe/amca). It has been designed to run fully in Python, and installation and operation can be performed easily and swiftly. Prior to the acquisition, the user defines the rough positions in which in the system scans for cells using the 'collect_position.py' script. With the script running, the user scans the slide manually and saves the positions of the stage at key points around the area to be imaged. A minimum of four points is required to scan a rectangular area. At each location, the user coarsely focuses the microscope on the cells using the z-piezo and stores the location. Once complete, the algorithm interpolates the positions across the entire area, with a user-defined sampling rate (e.g., every 200 µm xy, 0.5 µm z). This array of spatial locations forms the basis from which the acquisition of each stack/volume takes place in the xy dimension.

The acquisition is performed through a script called amca.py and it is designed to run in a Python 3.5+ environment. This script will signal the microscope to move around the slide either in the xy or z dimension by interfacing directly with the control hardware. In each location of interest, the script will signal the camera to acquire an image. The camera is controlled, and the image transferred from the camera, using a Python library written by Photometrics. The Photometrics library provides Python bindings specific for the Arch64 drivers, which can be provided on request from Photometrics (for library and drivers, see https://github.com/Photometrics/PyVCAM). The 'amca.py' script analyzes the camera image using the object detection algorithm of choice (typically YOLOv2). If any cells are detected in the image at this position, the microscope will be signaled to move up in the z dimension to the next position. An image is then acquired in this location. If again, cells are detected within the image, the image is saved and the microscope triggered to move up. This process is repeated until the focal plane has moved beyond the cells in this xy position and thus no more cells are detected. At this point the microscope is instructed to return to the initial z position first visited in this xy location and then to move down in the z dimension until again cells are no longer detected. The image stack is saved as a TIFF file either in the ImageJ or OME-TIFF standard and the ROI detected is embedded within the file's metadata. Next, the microscope is triggered to move and the process repeated at the next xy location defined earlier. The CoolLED lamp can either be left on throughout the experiment or triggered via Photometrics camera Python bindings. Furthermore, it is possible to interface directly with the CoolLED lamp via the USB and through using microscope control software written in Python (https://www.python-microscope.org/). Triggering is faster than USB control during acquisition, but for the triggering to work, the lamp must be preprogrammed (i.e., with exposures and illumination sequence) either via USB or manually through the external lamp control module. Once all the xy locations have been explored the system stops and notifies the user that the acquisition is complete. Image volumes are stored on a secure digital flash drive and can be then moved to another computer or uploaded to a server or OMERO system.

An important aspect of the AMCA is its ability to extract the cells from volumes once detected in the individual slices; one cell detected in one z-slice, is not by default connected by reference to the same cell detected in the following slice. This connectivity problem is nontrivial to solve, as the detected cell regions do not necessarily overlap perfectly between the slices and may appear intermittently if the detection accuracy is low. To solve this challenge, we modified a popular tracking algorithm called SORT (simple online and real-time tracker; Bewley, A., Z. Ge, L. Ott, F. Ramos, and B. Upcroft. 2016. 2016 IEEE International Conference. 3464–3468) software and used it for linking the object detections between 'z' slices to form volumetric bounding regions for each detected cells. Algorithms such as SORT ensure that an object labeled in one image is connected to the same-labeled object in a subsequent frame. SORT is based on the principle of a Kalman filter, which means that it can accommodate significant perturbations to the cells. Typically, SORT is applied offline, after acquisition, but this can be applied online also if needed. Once a cell's bounding volume has been uniquely identified, it is added to the TIFF file metadata along with the other ROIs.

Subsequent analysis of images and regions acquired using AMCA were performed using ImageJ/Fiji and Python scripts. For this project, scripts were developed that would allow access of datasets directly from an OMERO (Open Microscopy Environment - RO) instance or through processing of files in a folder located on a local machine. All these scripts, along with detailed instructions, are available in the repository (https://github.com/dwaithe/amca/tree/master/scripts).

Online supplemental material

Fig. S1 shows example data generated for study with corresponding ground-truth human annotations and object detection predictions. Fig. S2 shows the AP of Faster-RCNN, YOLOv2, YOLOv3, and RetinaNet at different levels of training for six independent datasets. Fig. S3 shows a comparison of object detection networks for cellular detection. Fig. S4 shows a summary performance of YOLOv2 when trained on multichannel data versus single-channel data. Table S1 shows a summary performance of Faster-RCNN, YOLOv2, YOLOv3, and RetinaNet networks on test datasets. Video 1 shows AR binocular in operation. Video 2 displays AR binocular in operation showing bounding boxes with cell intensity. Data S1 shows the schematics for the custom adapter plates and cube holder for the augmented reality modifications.

Acknowledgments

We thank John Prentice (Workshop Manager, University of Oxford, Oxford, UK) for his skilled technical and engineering input. We thank Ewan Mac Mahon (Systems Administrator, University of Oxford, Oxford, UK) for his excellent support of the computational resources used during this study. We also thank Silvia Galiani and Iztok Urbancic (Human Immunology Unit, University of Oxford, Oxford, UK) for their advice while setting up the microscope within the nanoimmunology laboratory, as well as the Wolfson Imaging Centre Oxford and Christoffer Lagerholm for general support. The Jetson TX2 development board used for this research was donated by the Nvidia Corporation, Santa Clara, CA.

We would like to acknowledge the UK Research and Innovation Biotechnology and Biological Sciences Research Council (BB/P026354/1) and the UK Research and Innovation Molecular Research Council (MR/S005382/1a, MC_UU_12009, MC_UU_12010/unit programs G0902418 and MC_UU_12025, MR/K01577X/1) for support of this project, as well as the Deutsche Forschungsgemeinschaft (Research Unit 1905 "Structure and function of the peroxisomal translocon," Jena Excellence Cluster "Balance of the Microverse," Collaborative Research Center 1278 "Polytarget"), the EPA Cephalosporin Fund, the Wellcome Trust (grant 104924/14/Z/14 and Strategic Award 091911 [Micron]), the Wolfson Foundation (for initial funding of the Wolfson Imaging Centre Oxford), and the John Fell Fund.

The authors declare no competing financial interests.

Author contributions: D. Waithe: conceptualization, data curation, formal analysis, funding acquisition, investigation, methodology, project administration, resources, software, validation, visualization, and writing (original draft, review and editing). J.M. Brown: c methodology, resources, and writing (review and editing). K. Reglinski: resources and writing (review and editing). I. Diez-Sevilla: resources. D. Roberts: supervision. C. Eggeling: supervision, funding acquisition, and writing (review and editing).

Submitted: 28 March 2019
Revised: 5 May 2020
Accepted: 21 July 2020

References

Allan, C., J.-M. Burel, J. Moore, C. Blackburn, M. Linkert, S. Loynton, D. Macdonald, W.J. Moore, C. Neves, A. Patterson, et al. 2012. OMERO: flexible, model-driven data management for experimental biology. *Nat. Methods*. 9:245–253. https://doi.org/10.1038/nmeth.1896

Andreopoulos, A., and J.K. Tsotsos. 2013. 50 years of object recognition: Directions forward. *Comput. Vis. Image Underst.* 117:827–891. https://doi.org/10.1016/j.cviu.2013.04.005

Bellomo, F., D.L. Medina, E. De Leo, A. Panarella, and F. Emma. 2017. High-content drug screening for rare diseases. *J. Inherit. Metab. Dis.* 40: 601–607. https://doi.org/10.1007/s10545-017-0055-1

Brown, J.M., N.A. Roberts, B. Graham, D. Waithe, C. Lagerholm, J.M. Telenius, S. De Ornellas, A.M. Oudelaar, C. Scott, I. Szczerbal, et al. 2018. A tissue-specific self-interacting chromatin domain forms independently of enhancer-promoter interactions. *Nat. Commun.* 9:3849. https://doi.org/10.1038/s41467-018-06248-4

Chen, P.C., K. Gadepalli, R. MacDonald, Y. Liu, S. Kadowaki, K. Nagpal, T. Kohlberger, J. Dean, G.S. Corrado, J.D. Hipp, et al. 2019. An augmented reality microscope with real-time artificial intelligence integration for cancer diagnosis. *Nat. Med.* 25:1453–1457. https://doi.org/10.1038/s41591-019-0539-7

Çiçek, Ö., A. Abdulkadir, S.S. Lienkamp, T. Brox, and O. Ronneberger. 2016. 3D U-Net: learning dense volumetric segmentation from sparse annotation. *In* Medical Image Computing and Computer-Assisted Intervention. Springer, New York. 424–432.

Dollár, P., C. Wojek, B. Schiele, and P. Perona. 2012. Pedestrian detection: an evaluation of the state of the art. *IEEE Trans. Pattern Anal. Mach. Intell.* 34:743–761. https://doi.org/10.1109/TPAMI.2011.155

Edwards, P.J., A.P. King, D.J. Hawkes, O. Fleig, C.R. Maurer, Jr., D.L. Hill, M.R. Fenlon, D.A. de Cunha, R.P. Gaston, S. Chandra, et al. 1999. Stereo augmented reality in the surgical microscope. *Stud. Health Technol. Inform.* 62:102–108.

Everingham, M., S.A. Eslami, L. Van Gool, C.K. Williams, J. Winn, and A. Zisserman. 2015. The pascal visual object classes challenge: A retrospective. *Int. J. Comput. Vis.* 111:98–136. https://doi.org/10.1007/s11263-014-0733-5

Girshick, R. 2015. Fast r-cnn. *arXiv*:1504.08083 (Preprint posted 30 April 2015)

Hollandi, R., A. Szkalisity, T. Toth, E. Tasnadi, C. Molnar, B. Mathe, I. Grexa, J. Molnar, A. Balind, and M. Gorbe. 2020. nucleAIzer: A parameter-free deep learning framework for nucleus segmentation using image style transfer. *Cell Syst.* 10:453–458.e6. https://doi.org/10.1016/j.cels.2020.04.003

LeCun, Y., Y. Bengio, and G. Hinton. 2015. Deep learning. *Nature*. 521: 436–444. https://doi.org/10.1038/nature14539

Li, J., and N.M. Allinson. 2008. A comprehensive review of current local features for computer vision. *Neurocomputing*. 71:1771–1787. https://doi.org/10.1016/j.neucom.2007.11.032

Li, Y., S. Wang, Q. Tian, and X. Ding. 2015. Feature representation for statistical-learning-based object detection: A review. *Pattern Recognit*. 48: 3542–3559. https://doi.org/10.1016/j.patcog.2015.04.018

Liu, W., D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A.C. Berg. 2016. Ssd: Single shot multibox detector. *In* European Conference on Computer Vision. Springer, New York. 21–37.

Pawlowski, N., J.C. Caicedo, S. Singh, A.E. Carpenter, and A. Storkey. 2016. Automating morphological profiling with generic deep convolutional networks. *bioRxiv*. doi:10.1101/085118 (Preprint posted November 2, 2016)

Redmon, J., and A. Farhadi. 2017. YOLO9000: better, faster, stronger. *arXiv*: 1612.08242 (Preprint posted December 25, 2016)

Redmon, J., and A. Farhadi. 2018. YOLOv3: An Incremental Improvement. *arXiv*:1804.02767 (Preprint posted April 8, 2018)

Ren, S., K. He, R. Girshick, and J. Sun. 2015. Faster r-cnn: Towards real-time object detection with region proposal networks. *In* Adv Neural Inf Process Syst. 91-99.

Ronneberger, O., P. Fischer, and T. Brox. 2015. U-net: Convolutional networks for biomedical image segmentation. *In* Medical Image Computing and Computer-Assisted Intervention. Springer, New York. 234–241.

Ruiz-del-Solar, J., P. Loncomilla, and N. Soto. 2018. A Survey on Deep Learning Methods for Robot Vision. *arXiv*:1803.10862 (Preprint posted March 28, 2018)

Schindelin, J., I. Arganda-Carreras, E. Frise, V. Kaynig, M. Longair, T. Pietzsch, S. Preibisch, C. Rueden, S. Saalfeld, B. Schmid, et al. 2012. Fiji: an open-source platform for biological-image analysis. *Nat. Methods*. 9: 676–682. https://doi.org/10.1038/nmeth.2019

Schmidt, U., M. Weigert, C. Broaddus, and G. Myers. 2018. Cell Detection with Star-convex Polygons. *arXiv*:1806.03535 (Preprint posted June 9, 2018).

Stanley, W.A., F.V. Filipp, P. Kursula, N. Schüller, R. Erdmann, W. Schliebs, M. Sattler, and M. Wilmanns. 2006. Recognition of a functional peroxisome type 1 target by the dynamic import receptor pex5p. *Mol. Cell*. 24:653–663. https://doi.org/10.1016/j.molcel.2006.10.024

Stringer, C., M. Michaelos, and M. Pachitariu. 2020. Cellpose: a generalist algorithm for cellular segmentation. *bioRxiv*.

Suleymanova, I., T. Balassa, S. Tripathi, C. Molnar, M. Saarma, Y. Sidorova, and P. Horvath. 2018. A deep convolutional neural network approach for astrocyte detection. *Sci. Rep*. 8:12878. https://doi.org/10.1038/s41598-018-31284-x

Sun, Z., G. Bebis, and R. Miller. 2006. On-road vehicle detection: a review. *IEEE Trans. Pattern Anal. Mach. Intell.* 28:694–711. https://doi.org/10.1109/TPAMI.2006.104

Thomas, N.. 2010. High-content screening: a decade of evolution. *J. Biomol. Screen.* 15:1–9. https://doi.org/10.1177/1087057109353790

Verschae, R., and J. Ruiz-del-Solar. 2015. Object detection: current and future directions. *Front. Robot. AI.* 2:29. https://doi.org/10.3389/frobt.2015.00029

Weigert, M., U. Schmidt, T. Boothe, M. Andreas, A. Dibrov, A. Jain, B. Wilhelm, D. Schmidt, C. Broaddus, and S. Culley. 2017. Content-aware image restoration: pushing the limits of fluorescence microscopy. *Nat Methods*. 15:1090–1097.

Yu, W., H.K. Lee, S. Hariharan, W. Bu, and S. Ahmed. 2010. Evolving generalized Voronoi diagrams for accurate cellular image segmentation. *Cytometry A*. 77:379–386. https://doi.org/10.1002/cyto.a.20876

Supplemental material

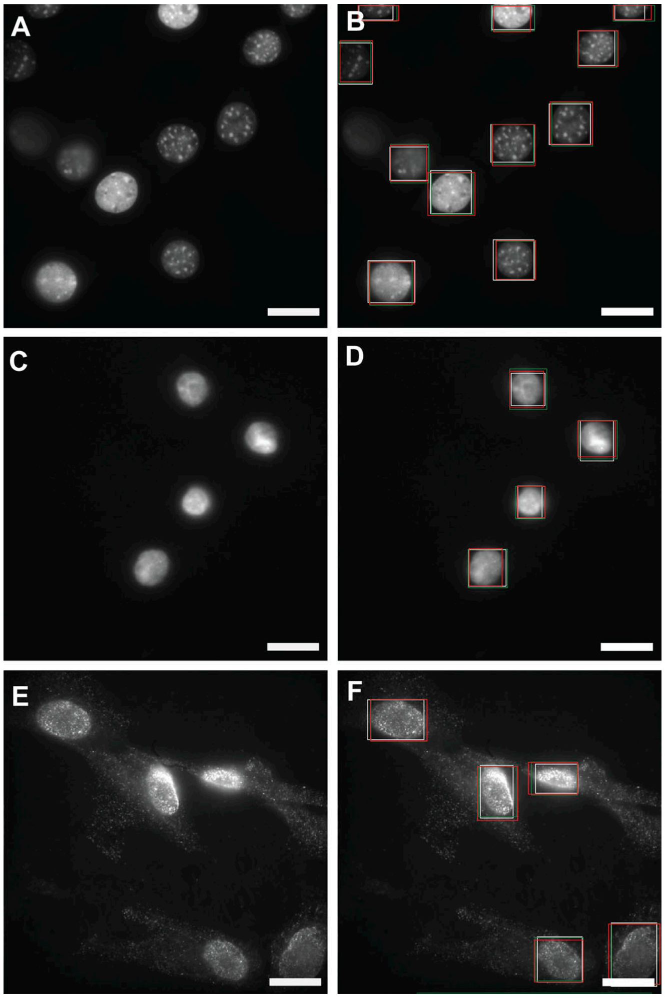

Figure S1. **Example data generated for study with corresponding ground-truth human annotations and object detection predictions. (A and B)** C127 cell dataset, stained with DAPI. **(C and D)** Erythroblast cells stained with DAPI. **(E and F)** Fibroblast cells stained for a nucleopore protein. Ground-truth boxes, (white), YOLOv2 prediction boxes (red), and Faster-RCNN prediction boxes (green). Scale bars, 25 µm.

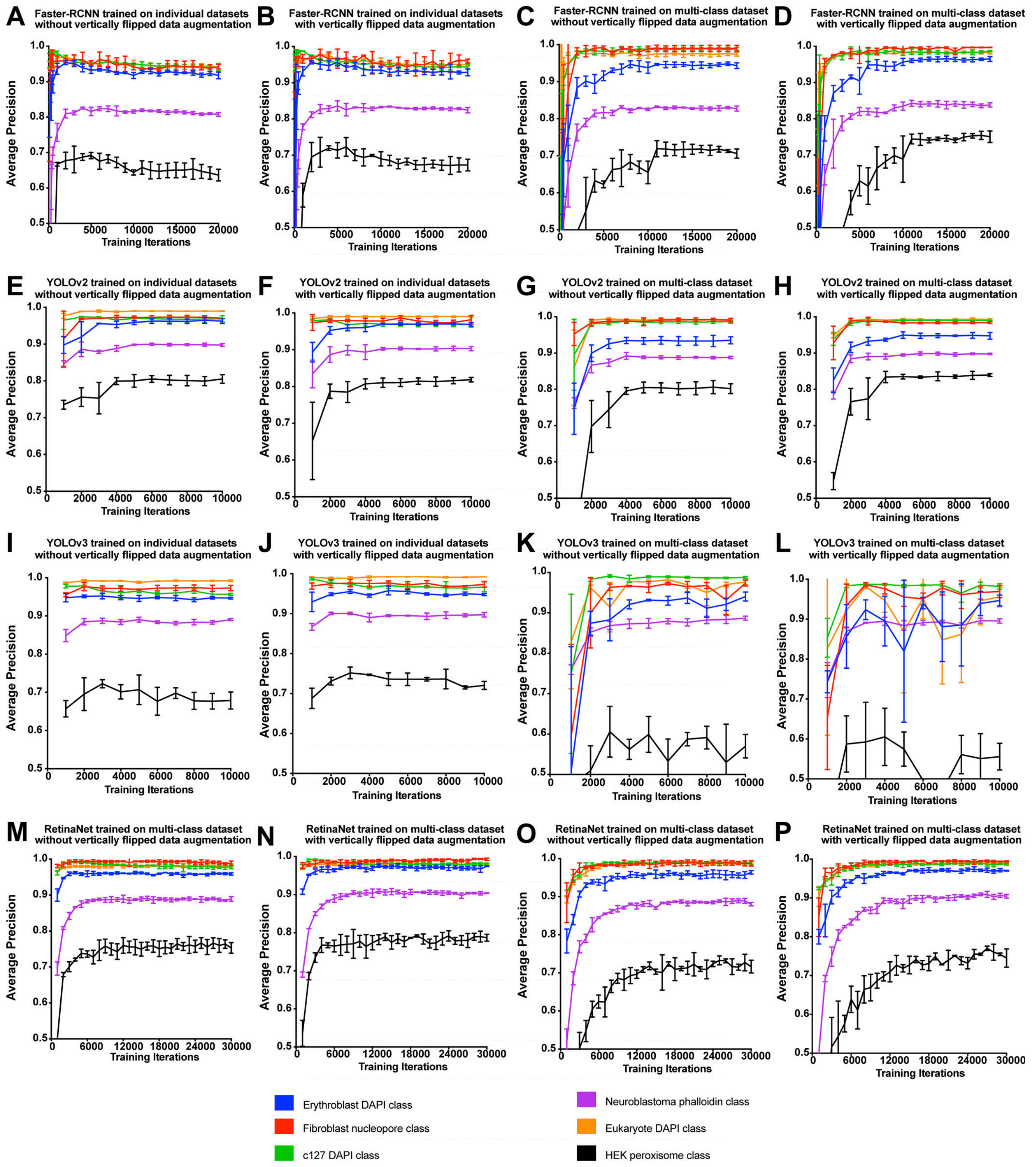

Figure S2. **AP of Faster-RCNN, YOLOv2, YOLOv3, and RetinaNet at different levels of training for six independent datasets.** AP for each algorithm trained and evaluated on individual datasets without (TR1; A, E, I, and M) and with (TR2; B, F, J, and N) additional vertically flipped training data augmentation. AP for algorithm trained across multiple datasets and evaluated on individual datasets without (TR3; C, G, K, and O) and with (TR4; D, H, L, and P) additional vertically flipped training data augmentation. Erythroblast DAPI cells (blue), neuroblastoma phalloidin dataset (magenta), fibroblast nucleopore dataset (red), eukaryote DAPI dataset (orange), C127 DAPI dataset (green), and HEK peroxisome dataset (black). Error bars (*n* = 3, AP ± SD).

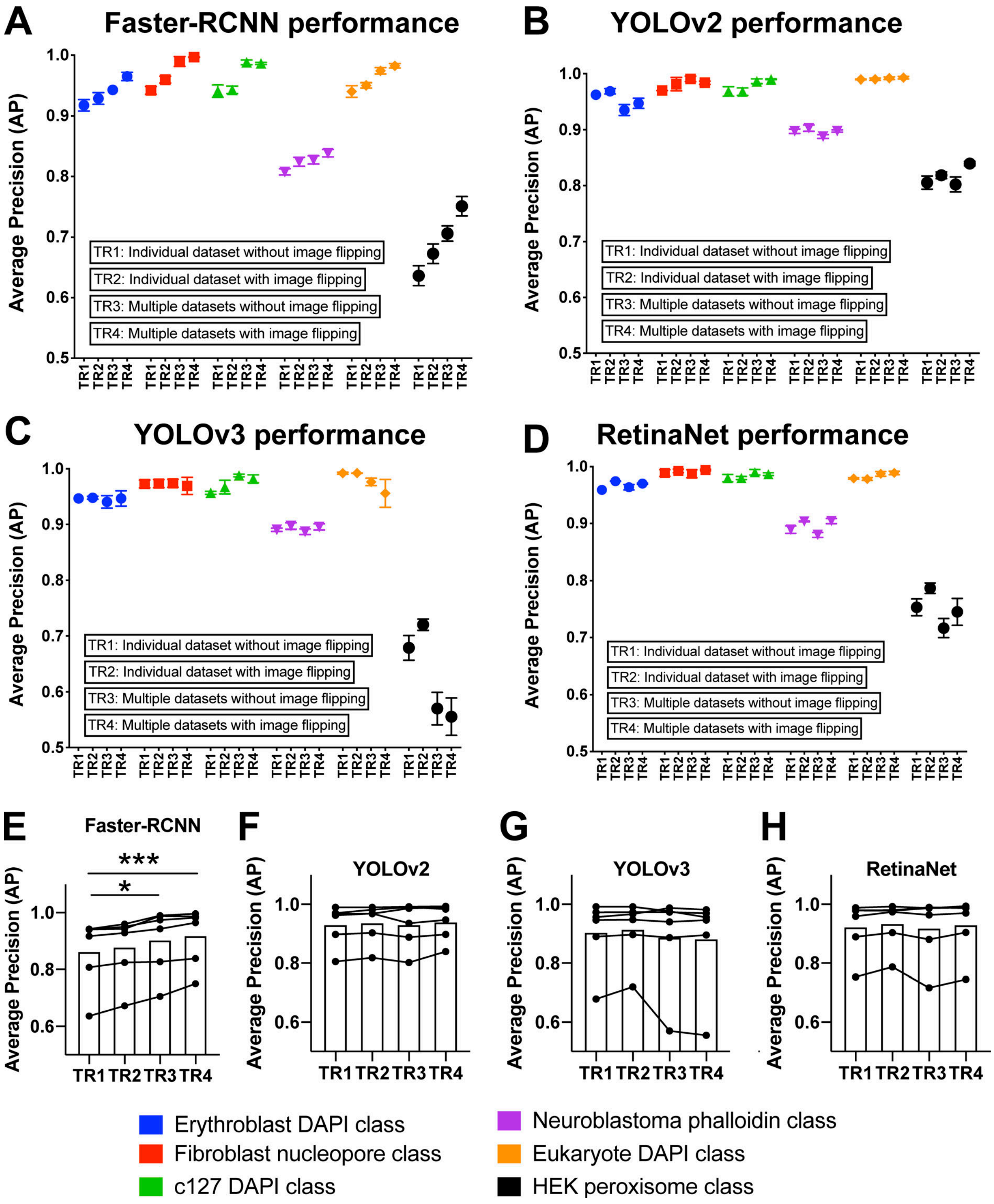

Figure S3. **Comparison of object detection networks for cellular detection.** Performance in terms of AP of Faster-RCNN (A), YOLOv2 (B), YOLOv3 (C), and RetinaNet (D) when trained on individual datasets without (TR1) and with vertically flipped data augmentation (TR2) and when trained using multiple datasets without (TR3) and with (TR4) vertically flipped data augmentation (*n* = 3, AP ± SD). Erythroblast DAPI cells (blue), Neuroblastoma phalloidin dataset (magenta), fibroblast nucleopore dataset (red), eukaryote DAPI dataset (orange), C127 DAPI dataset (green), and HEK peroxisome dataset (black). AP performance across all datasets for Faster-RCNN (E), YOLOv2 (F), YOLOv3 (G), and RetinaNet (H) for TR1–TR4. Additional vertical flipping of data (TR2) and joint training with multiple classes statistically boosts AP when using the Faster-RCNN networks, but not the other networks. Friedman's test was applied using Dunn's multiple comparison test to compare TR2–TR4 to T1 (*n* = 6, AP ± SD; *, P < 0.05; ***, P < 0.005).

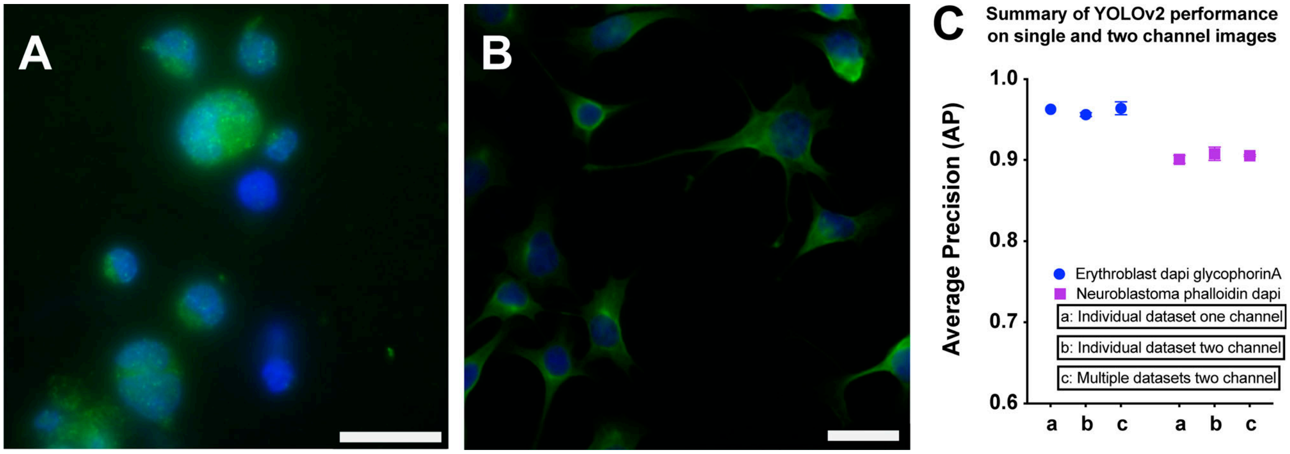

Figure S4. **Summary performance of YOLOv2 when trained on multichannel data versus single-channel data. (A)** Erythroblast cells stained with DAPI (blue) and for glycophorin A protein (green). **(B)** Neuroblastoma cells stained with phalloidin (green) and DAPI (blue). Scale bars represent 25 µm in both images. All training material includes vertically flipped data augmentation, and the ground-true for the first channel (DAPI) was accessed in both cases. **(C)** AP of YOLOv2, comparing performance when trained and evaluated on a single dataset comprising one-channel data (a) and two-channel data (b) or when trained on multiple data comprising two channels (c). Erythroblast DAPI glycophorin A dataset (blue) and neuroblastoma phalloidin DAPI dataset (magenta; *n* = 3, mean ± SD).

Video 1. **Augmented reality demonstration video.** Video was acquired using iPhone SE camera positioned at the binocular eye-piece. The frame-rate is 15 frames per second..

Video 2. **Augmented reality demonstration video showing bounding boxes with intensity graded to cellular intensity.** Video was acquired using iPhone SE camera positioned at the binocular eye-piece. The frame-rate is 15 frames per second..

Table S1 is provided online and shows summary performance of Faster-RCNN, YOLOv2, YOLOv3, and RetinaNet networks on test datasets. Data S1 shows the schematics for the custom adapter plates and cube holder for the augmented reality modifications.