# CENTRACK: A deep learning system for scoring centrioles in wide-field fluorescence microscopy

Leo Bürgy, Martin Weigert, Matthias Minder, Georgios Hatzopoulos, Gabriela Garcia-Rodriguez, Keshav Jha, Pierre Gönczy

# Main

## Context

Centrioles are organelles that play a central role in the cell cycle. Their numbers vary normally between 2 and 4 in human cells. Their molecular structure consists of hundreds of proteins, many of which remained undefined in their identity and localisation. Some changes in their components can trigger abnormalities in their duplication. Therefore, the study of their constituents often involves at some point the analysis of the centrioles from microscopy data, it being the intensity of the signal using a particular marker or simply the number of centrioles per cell.

The detection of centrioles and its complementary task of scoring cells for centrioles are central for characterising the phenotype of cell populations. Thus far, it has remained a manual task at the microscope. This routine is slow, tedious, error prone and not reproducible. As there is no image acquisition, it is impractical, if not impossible, to label already scored cells. Moreover, training in any sense is impossible—new experimenter cannot learn from labelled field of view and potential images cannot be used for learning based method.

Attempts to automate the detection/scoring exist but at the centrosome level. The more classical way is to model a one-channel image as a set of pixels, where centrioles are brighter by an order of magnitude compared to the background. Assuming that all centriole pixels are in the same intensity range, one can determine a threshold value, to pick out the centriole positions. Unfortunately, this approach is not robust enough as intensity varies too much between centrioles and between cells. The more sophisticated method uses deep learning.

Besides that, detecting pairs of centrioles (~centrosomes) is a much simpler task as the clustered centrioles do not need to be kept separated. Another challenge in this detection task is the nested structure of the data to be detected. In brief, field of view contain cells that contain centrosome that contain mother centrioles that may contain daughter centrioles. This is challenged by the fact that some markers of centrioles appear in either one or both centriole type.

The advantage of automatic annotation is undeniable: it allows reproducible and persistent knowledge. In this context, automating the detection of the foci represents a good alternative. Several attempts have been done to detect bright spots of a certain type in immunofluorescence images either using classical computer vision [REF] or using convolution neural networks [REF]. Here we present centrack, a system for annotating cells and their centrioles in immunofluorescence (IF) data. We also present the workhorse of the pipeline, a variant of U-Net that tackle the high resolution/small SOI size challenge.

## Modalities

The core data types for centrack are multimodal fluorescence imaging of cells in vitro or in vivo. The data are in the bio-formats/OME-TIFF formats and are of the form Channel-Depth-Width-Height, or for short, CZYX. The channels usually include a marker for nuclei and several markers of centrioles. Stacks are acquired along Z so that cells are imaged in their entirety. However, it is customary to reduce their dimensionality with a max-projection along the Z-dimension. Y and X for now need to be equal to 2048.

## Datasets

To show the application of centrack, we used three different datasets (DS) that share common structure but are different in their end goals. The first dataset **(cells)** contains 25 field of view of cycling human retinal pigment epithelial cells (hTERT RPE-1) fixed with methanol and immuno-stained. Nuclei (blue) are stained with Hoechst (blue), while bright spots mark the centrioles stained with Centrin-2 (magenta) and Cep63 (green). Image was acquired on a Zeiss Observer D1 with a 63x oil immersion objective and an Andor Zyla 4.2p. The image stack has been max projected along the z-axis. This kind of dataset is representative of ubiquitous routines in the field, where the assignment is important to properly score centrioles per cell. One difficulty is the difference in brightness and the spurious blobs that can be taken as centrioles. A second difficulty is that the foci can be only a few pixels apart. But no compromise can be tolerated as this telling apart is the end goal of the task. Third, some cells can be dividing and depending on the stage, it can be challenging to properly assign centrioles to either cell/nucleus. Fourth, some nuclei or centrioles may be only partially visible in the FOV. These should be excluded from the detection set. The second dataset (**denovo**) contains human cells whose centriole formation has been exacerbated. Therefore, the number of centrioles in each cell can be high. In addition, the foci can be more difficult to determine manually. This poses the challenge that the ground truth may be of lesser quality. The third dataset (**gonad**) contain gonads of C. elegans only labelled for one centriole marker (SAS-7). In this dataset, all foci of a predefined region must be detected and the intensity of the signal must be measured in order to compare it with other treatments. Therefore, the assignment is not necessary.

## Architecture

We designed centrack to be used for various tasks. Centrack detects all the foci present in a field of view. We strived to keep the foci detection separate from the assignment. Only in a second time are the foci assigned to nuclei. This is because sometimes, the foci only need to be detected irrespective of their ontology (DS3).

Centrack consists of base classes to model the different objects it operates on, such as the image data (Dataset, Stack, Channel), the structures of interest (SOI) to detect (Centriole, Centrosome, Nucleus, Cell) as well as Detector that run a model on Channel to output a set of SOIs. The API exposes the user to the following entry points: (1) **preparation** of the stacks, essentially the projection type (mean, max, sum); (2) the **detection** step where the detection is taking place; (3) the **measurements** that are carried out on the detected SOIs. Besides this canonical workflow. We also provide utilities to check the structure of the dataset, to create vignettes to be uploaded to an annotation tool, to check config files generated by the user.

Three types of measurements relate to centriole detection: (1) Scoring cells for **centriole number**; (2) Sampling centriole **intensity**; (3) **Comparing** presence across channels.

## Models

Centriole are detected from single max projected channels and their coordinates are recorded with respect to the original projection resolution, that is 2048x2048. The model was trained on the dataset **cells**. If the assignment is done using the nearest-nucleus method, the nuclei are segmented using the pretrained StarDist model. Then, each nucleus is converted into a OpenCV contour using the function findContours from OpenCV. For each nucleus, all detected centrioles within a given distance is assigned to it. Then the nucleus is removed from the list to assign until all nuclei have been evaluated. The limitations of this method are clear: if two cells are confluent, there is a risk that foci may be assigned to the other cell. Also, the distance is hard coded and does not scale well. As an alternative method we allow the user to extract the extent of the cells using the same channel as the one used for centriole detection. This relies on the fact that some centriolar component are also present in the cytoplasm. This approach has the advantage of not depending on distance-based metrics. The assignment is then read out by retrieving the pixel value at the focus location in the segmentation mask where each cell is labelled with its index. The limitation of the second approach is the dependency on the quality of the signal.

*Figure 1: Overview of the pipeline and architecture of SpotNet. A. The z-stack is z-projected and the channel of interest passed to the model for centriole detection. B. SpotNet has a U-Net backbone that consists of a encoder, followed by a bottleneck and a decoder. The model takes as input a YX image. It is first normalised and passed through three convolving blocks.*

*Figure 2: Examples applications of the pipeline in centriole analyses. A: Regular scoring involves the detection of centrioles followed by the assignment to their nearest nucleus. B: de novo centrioles are less clear cut to segment. Thus, the ground truth annotation may be more challenging as many more spots are going to be considered as centrioles. C:*

# Methods

## Datasets (~5 x 30)

1. RPE cells, marker set 1: Image of cycling human retinal pigment epithelial cells (hTERT RPE-1) fixed with MetOH and immuno-stained. Nuclei (blue) are stained with Hoechst (blue), while bright spots mark the centrioles stained with Centrin-2 (magenta) and Cep63 (green). Image was acquired on a Zeiss Observer D1 with a 63x oil immersion objective and an Andor Zyla 4.2p. The image stack has been max projected along the z-axis.
2. RPE cells, marker set 2
3. RPE cells, marker set 3
4. *de novo* centrioles
5. C. elegans ovaria
6. C. elegans embryo

## Software stack

The whole pipeline is implemented in Python version 3.9.5. It was tested using pytest and the dependency as well as the packaging was done using poetry and is available from pypi.org for download. The models were implemented in TensorFlow 2.0. The nuclei were segmented using the DAPI channel max-projected and binned so that the input of StarDist is 512x512.

## Labelling workflow

Manual annotation to generate the ground truth was done with Labelbox interface. The projection files were contrast adjusted and converted in PNG format and uploaded using a Python script (upload\_labelbox.py). In Labelbox, the ObjectAnnotation were Points for foci and Freehand segmentation for cells. The pretrained model for nuclei segmentation were used as is as visual inspection of the prediction gave no sign of misfit.

Network architecture

The model was implemented in Tensorflow 2.0 using backbone code from CSBDEEP library. It consists of a multiscale U-Net based architecture with 3 convolution blocks (Conv, BatchNorm, Maxpool, Conv, Maxpool), which each has a kernel size of 3x3. The high resolution of the input (2048x2048) imposes that it needs to be fed by patches of 32x32.

## Model training and testing

The model was trained on DS1 for x epochs with a learning rate of Y on a workstation Z.

## Model evaluation

As we decoupled the detection of foci from the proper scoring, we could concentrate the attention on the accuracy of the focus detection. We made use of the linear assignment between the foci annotation set (the ground truth) and the foci prediction set. We used the Hungarian algorithm implemented in Python with the Euclidean distance as the objective function.

Assignment

The assignment of the foci to cells, that is, the scoring was done using two different approaches. In the first approach, the nuclei are detected and for each nucleus, the foci with the minimal distance to the nucleus under consideration was chosen. We used the OpenCV function pointPolygonTest, which performs a point in contour test. We set the option to retain the signed distance of the point to the contour. That is that the distance is negative if the foci is outside of the contour. By computing the signed distance to all available nuclei, one can assign the focus to the nearest nucleus by taking the maximum signed distance.