nature portfolio

https://doi.org/10.1038/s41596-021-00576-4

Supplementary information

High-speed laser-scanning biological microscopy using FACED

In the format provided by the authors and unedited

High-speed laser-scanning biological microscopy using FACED

Queenie T. K. Lai[1,*], Gwinky G. K. Yip[1,*], Jianglai Wu[2,*] , Justin S. J. Wong[1], Michelle C. K. Lo[1], Kelvin C. M. Lee[1], Tony T.H.D. Le[1], Hayden K. H. So[1], Na Ji[2,3,4,5], and Kevin K. Tsia[1,6]

1Department of Electrical and Electronic Engineering, The University of Hong Kong, Hong Kong, China
2 Department of Physics, University of California, Berkeley, CA, USA
3 Department of Molecular and Cell Biology, University of California, Berkeley, CA, USA
4 Helen Wills Neuroscience Institute, University of California, Berkeley, CA, USA
5 Molecular Biophysics and Integrated Bioimaging Division, Lawrence Berkeley National Laboratory, Berkeley, CA, USA
6 Advanced Biomedical Instrumentation Centre, Hong Kong Science Park, Shatin, New Territories, Hong Kong
* These authors contributed equally to this work
e-mail: tsia@hku.hk; jina@berkeley.edu

Supplementary Manuals:

Manual 1	Active feedback stabilization loop for the FACED mirror pair
Manual 2	Detailed procedures in Step 39: Initial setup of the line clock generator
Manual 3	FPGA data acquisition logic and image processing pipeline (Step 41)

Supplementary Notes:

Note 1	Extracted biophysical phenotypes derived from FACED images

Supplementary Tables:

Table S1	List of all extracted biophysical phenotypes and the corresponding equations
Table S2	List of Variables Used in the Equations (Table S1)

Supplementary Figures

Figure S1	Technical drawings of the wafer of making the microfluidic chips.
Figure S2.	Different misalignment scenarios of the FACED mirror pair.
Figure S3.	The schematic diagram of the feedback stabilizer for FACED module
Figure S4.	The optical setup of the active feedback stabilizer system.
Figure S5.	PCB design of the analogue circuitries for signal conditioning.
Figure S6.	Comparison of the mechanical stability of the FACED mirror pair.
Figure S7.	Synchronized sample clock generation setup.
Figure S8.	Line clock generator control software serial connection to host PC
Figure S9.	Line clock generator control software configuration procedure.
Figure S10.	Control software for Digitizer sample clock multiplier (Valon 5009)
Figure S11.	ADQ7 input connections
Figure S12.	ADQ7 Digitizer hardware and FPGA programmable function layout
Figure S13.	Line and frame synchronization signals in FPGA User Logic 2 module.
Figure S14.	General schematic of the pixel-gradient-based cell region/segment detection hardware on FPGA for real-time data reduction.
Figure S15.	Architect of NN-based classification model and CNN-based regression model.
Figure S16.	Results of predicting the DNA content.
Figure S17.	Time-gated photon counting detection in FACED SHG imaging.

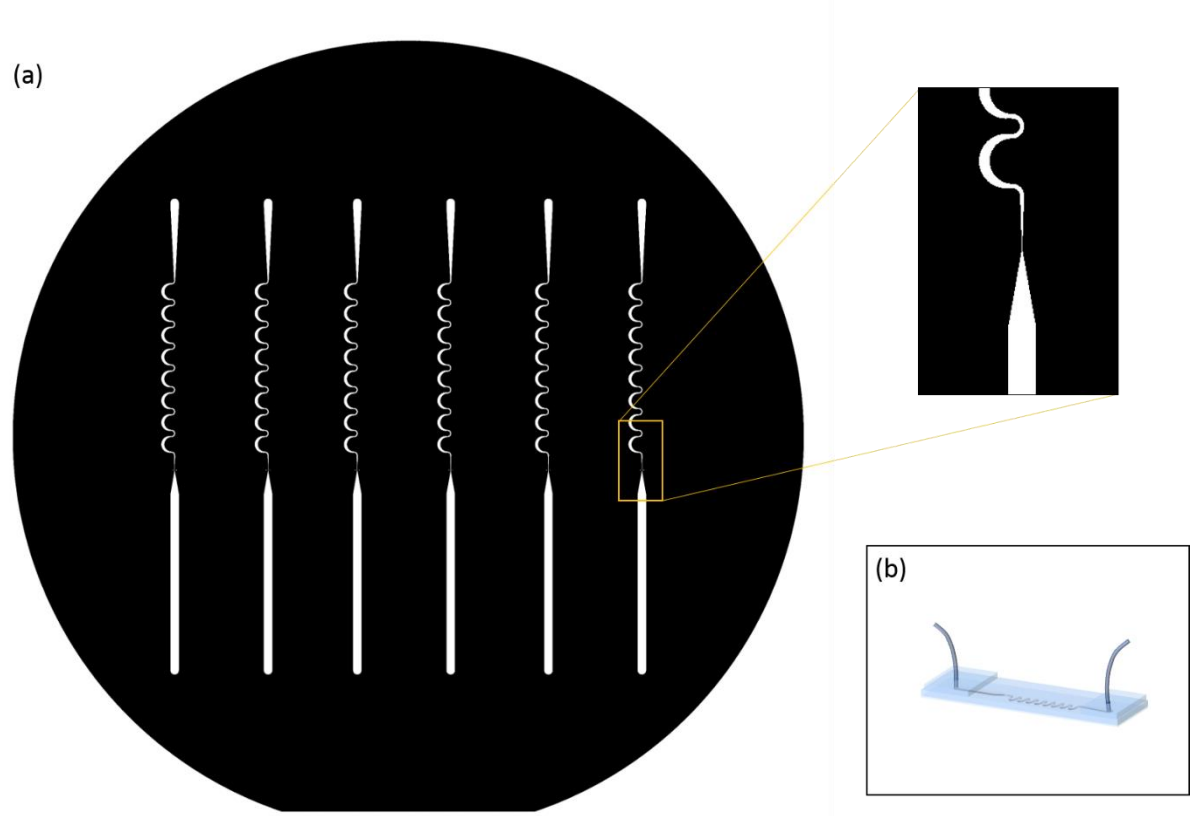

Figure S1 Technical drawings of the microfluidic chips. (a)The premade custom design silicon wafer mold for the microfluidic chips used in imaging flow cytometry experiments. (b) Microfluidic channel used in cell flowing imaging.

Supplementary Manual 1: Active feedback stabilization loop for the FACED mirror pair

Ideally, as the surfaces of the mirror pair and the propagating light are perpendicular and parallel to the optical table respectively, the incidence angle of the propagating light within the FACED mirror is always zero along the vertical direction (i.e. y-axis). However, due to mechanical instability, relative tilting between the mirror pair along the pitch axis always happens in practical setting (**Fig. S2**).

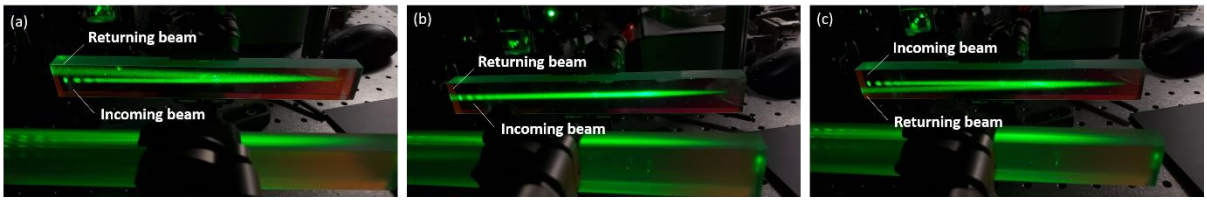

Figure S2. Different misalignment scenarios of the FACED mirror pair. (a) Tilted upward. (b) Normal. (c). Tilted downward.

Importantly, because of the long propagation distance between the mirror pair, the light propagation is sensitive to the mechanical drift of the mirror pair. The relationship between the total displacement of light beam *s* along the y-axis and the relative tilt angle β is shown as:

$$s = \beta S + 2\beta S + \cdots + 2N\beta S = (2N+1)N\beta S$$

where S is the separation between the mirror pair and *N* is the number of virtual sources.

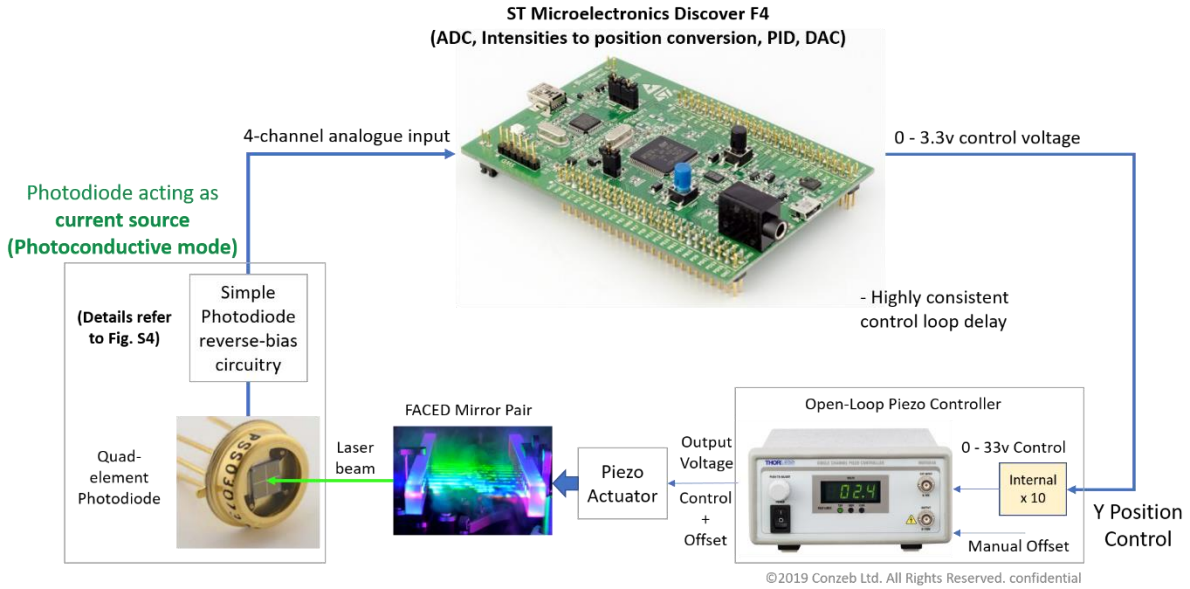

Figure S3. The schematic diagram of the feedback stabilizer for the FACED module.

The displacement of beam not only produces optical aberration, but also increases power loss due to leakage of light if the displacement is large. In order to maintain the imaging quality, an active feedback control system is developed to control the tilting of one of the mirrors (**Fig. S3).**

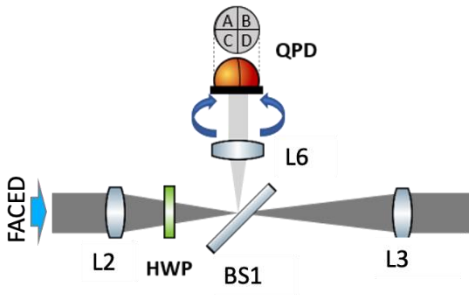

Figure S4. The optical setup of the active feedback stabilizer system.

Firstly, a pellicle beam splitter (Thorlabs) is placed after the mirror pair to split a portion of light to a quadrant photodetector (QPD) (First Sensor), which is located at the Fourier plane to detect the beam displacement along the vertical direction (**Fig. S4**). When one of the mirrors is tilted upward, the beam at Fourier plane will move upward, and vice versa. It is important to note that a pair of convex lens is used to compress the beam size to be within the sensor area. Next, the signals collected by the QPD are sent to the motherboard (ST Microelectronics Discovery F407) after signal conditioning by the home-made analogue circuitries (**Fig. S5**). The advantage here is that no amplification of signal is required and low electrical noise. In addition, the recovery time can be flexibly adjusted by changing the current-limiting resistor. Then, based on the calculated difference in collected power between the top and the bottom quadrants, the situation of tilting upward and downward can be identified and compensated by PID control. After that, a voltage signal is sent to the piezo controller (Thorlabs) to drive the piezo actuator of a custom-made mounting platform (Thorlabs). One of the mirrors is mounted on the platform with one piezo actuator to control the pitch of the platform. Finally, the mounted mirror is tilted so as to make β become zero.

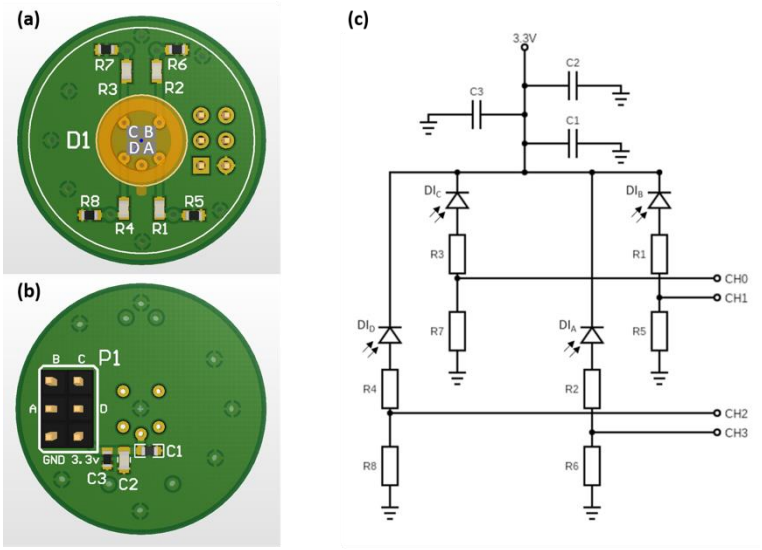

Figure S5. PCB design of the analogue circuitries for signal conditioning. (a) Front view of the PCB board. (b) Back view of the PCB board. (c) Schematic diagram of the PCB. CH0, CH1, CH2 and CH3 are the 4-channel analogue outputs from the PCB into the motherboard for further processing. R1 = R2 = R3 = R4 = 2.7kΩ, R5 = R6 = R7 =R8 = 10kΩ, C1 = 1nF, C2 = 10nF, C3 = 10μF and DI_x refers to the photodiode at the corresponding quadrant (i.e. A, B, C and D).

Without enabling the stabilizer, there are not only high frequency movements of the mirror due to mechanical vibration from the surrounding, but also long-term drift to tilt downward due to the gravity. By enabling the stabilizer, rapid fluctuation of beam displacement is minimized and the beam position is consistently maintained at a targeted height (**Fig. S6**).

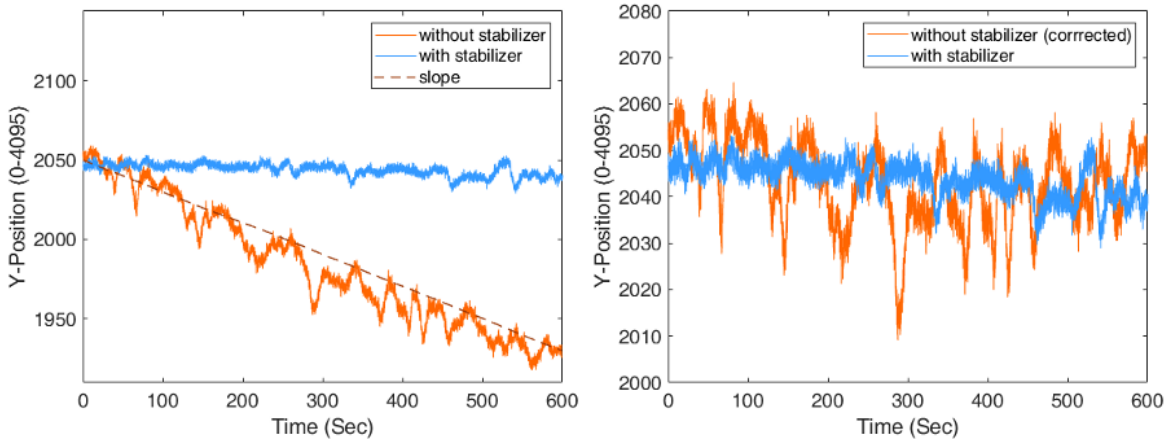

Figure S6. Comparison of the mechanical stability of the mirror pair between (blue) with and (orange) without enabling the stabilizer over 10 minutes of (left) raw signal and (right) corrected signal of the Y-position logger that is the raw signal minus the slope fitted.

Supplementary Manual 2 Detailed setup procedures of the line clock generator

Detailed procedures in Step 39: Initial setup of the line clock generator

Perform the initial setup of the line clock generator, which consists of a pulse delay generator (Aerodiode) with a photodiode module. It include the following procedures:

1. Connect the reference laser pulses via optical fibre to the photodiode input of the **Pulse Delay Generator (Fig. S7**), then connect it to the host PC via the provided USB cable.

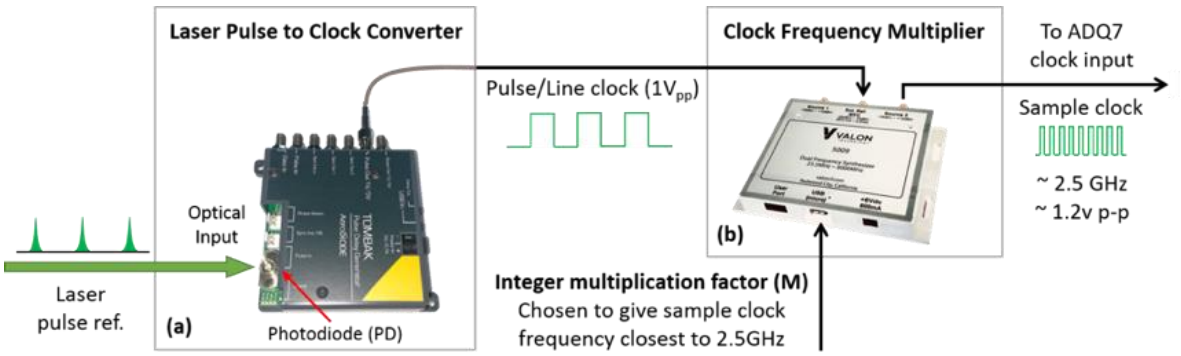

Figure S7 Synchronized sample clock generation setup.

2. Connect the 5v DC power supply to the **Pulse Delay Generator** to power on the device.
3. Start the provided control software on the host PC, then establish the serial connection to the **Pulse Delay Generator** by clicking on the **connect** button (**Fig. S8)** When connection is successful, the control graphical user interface (GUI) will appear (**Fig. S8**). [**Troubleshoot**: In the case of the software getting stuck completely on the initial connection, terminate the software completely, then retry until connection is successful.]

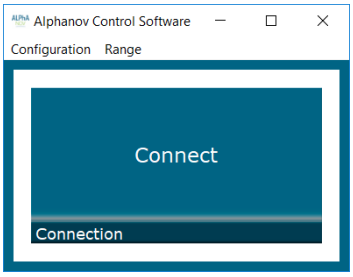

Figure S8 Line clock generator control software serial connection to host PC (*procedure)

4. Set the **threshold** voltage to 0.05v (**Fig. S9**(c)). Make sure all the options in the **Advanced Mode** settings (**Fig. S9(**b)) remain unselected (not highlighted), then set the input source to **Phot.** (Photodiode) in (**Fig. S9** (e)).
5. Check that the detected pulse frequency in (**Fig. S9**(d)) matches the expected laser pulse frequency.
   ?Troubleshooting: if the frequency is incorrect, try modifying the threshold voltage until the pulse frequency is correct
6. Set the **Output Pulse Delay** (**Fig. S9**(g)) to the minimum value (72.00ns), then disable **Auto Fine Delay** and set the value to 0.00ns (**Fig. S9**(h)).
7. Check that the **Pulse Out** SMA signal is configure on the hardware to 1v range.
8. Enable the **Pulse Out** signal by selecting **On** in the Board section (**Fig. S9**(a)). Check the actual output waveform to ensure that it has the correct range of 1v and has the correct frequency.
9. Adjust the **Output Pulse Width** (**Fig. S9**(f)) to ensure that the output waveform has approximately a 50% duty-cycle.
10. All other settings should be kept at their default values as shown in **Fig. S9.**
11. Once the output waveform is correct, the user may save the settings to file for quicker start-up in future use.

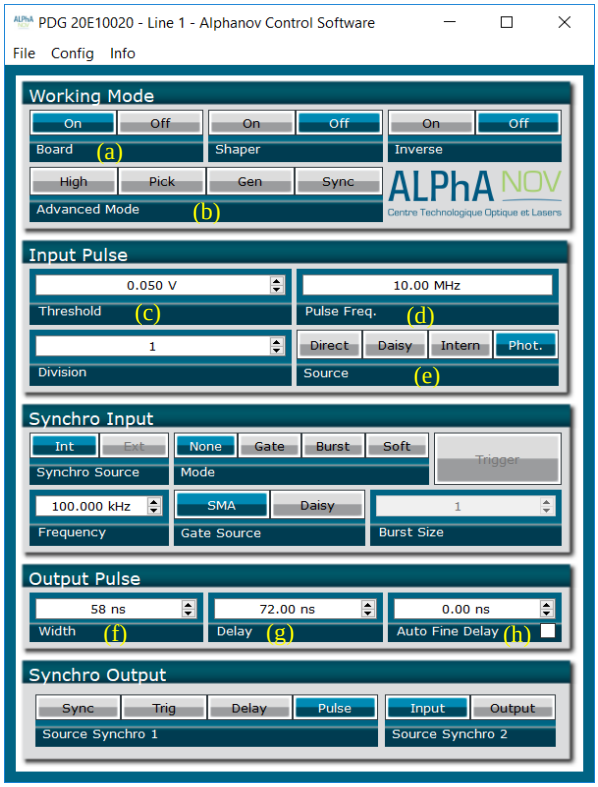

Figure S9 Line clock generator control software configuration procedure.

Detailed procedures in Step 39 : Initial setup of sample clock frequency multiplier (model: Valon 5009 clock synthesizer + PS6V-1 Power Supply Kit)

1. Connect the **Pulse Out** signal from the **Pulse Delay Generator** via SMA cable to the **Ext. Ref. input** of the **Valon 5009 clock synthesizer** (**Fig. S7**).
2. Power on the **Valon 5009 clock synthesizer** with the 6v DC 1A power supply.
3. Connect the **Valon 5009 to the host PC iva the provided USB cable.**
4. Run the Valon 5009 Configuration Manager software on the host PC, then set the **Reference Source** to **external** (**Fig. S10**)(d)).
5. Next, set the **Reference Freq** (**Fig. S10**(e)) to the laser pulse frequency. Note that this value is used mainly to ensure that the internal Phase-lock-loop (PLL) is configured with the correct settings and bandwidth limits, so a slight inaccuracy of the entered value will not matter.
6. Set the multiplying factor to INT (integer) (**Fig. S10**(b)).
7. Compute the integer frequency multiplying factor **M** according to **M = Round (2.5GHz / f_{pulse})**, where f_{pulse} should match exactly the **Reference Freq** value entered in step 5. Then use the resultant **M** to derive the actual **ADC clock frequency** using $f_{ADC} = M \times f_{pulse}$.
8. Enter the derived **ADC clock frequency** as the **Frequency** entry in **Fig. S10**(a).
9. Set the **Attenuator** setting (**Fig. S10**(c)) to 16.5 to ensure that the amplitude of the generated signal is within the safe limit of the clock input on the ADQ7 Digitizer.
10. Finally, check that the PLL (**Fig. S10** (f)) has successfully locked to the correct output frequency.
11. For quick configuration in subsequence start-ups, user can save the settings to a file.

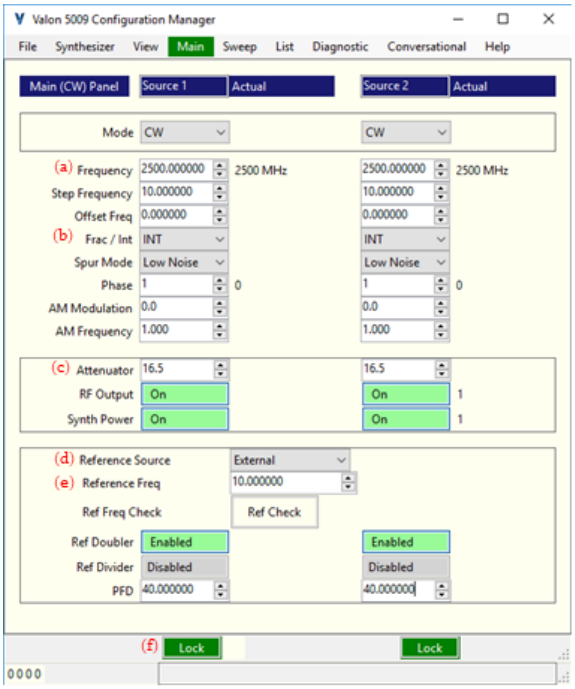

Figure S10 Control software for Digitizer sample clock multiplier (Valon 5009)

? Troubleshooting: If the Valon 5009 module fails to lock or the lock is not stable (blue LED indicator flashes or varying brightness). Please check that the reference clock frequency is configured correctly in the configuration software, and that the actual waveform of the input line clock has a stable/correct waveform, correct frequency, 50% duty-cycle, and 1v amplitude. If not, tune and correct the settings of the previous line clock generation stage.

Detailed procedures in Step 40 :Digitizer setup (Teledyne SP Devices ADQ7DC-PCIe)

1. Connect the **clock signal** from either **Source 1** or **Source 2** output of the **Valon 5009 clock synthesizer to** the **CLK** input of **ADQ7** (**Fig. S11)**
2. Connect the PD or PMT signal to the Input X on the ADQ7.
   CRITICAL STEP: The voltage level of the signal must be checked and limited to within +/- 1v using attenuator to prevent damaging the sensitive ADCs in ADQ7.

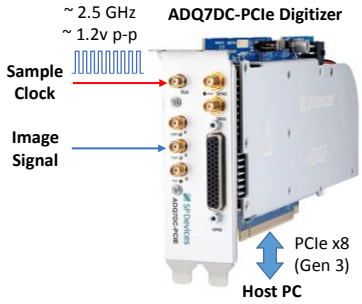

Figure S11 ADQ7 input connections

Supplementary Manual 3: FPGA data acquisition logic and image processing pipeline (Step 41)

The overall layout of the ADQ7 and its built in-logic and user customizable logic is shown in the **Fig. S12**. We will describe different ways of programming the User Logic 2 in this Note.

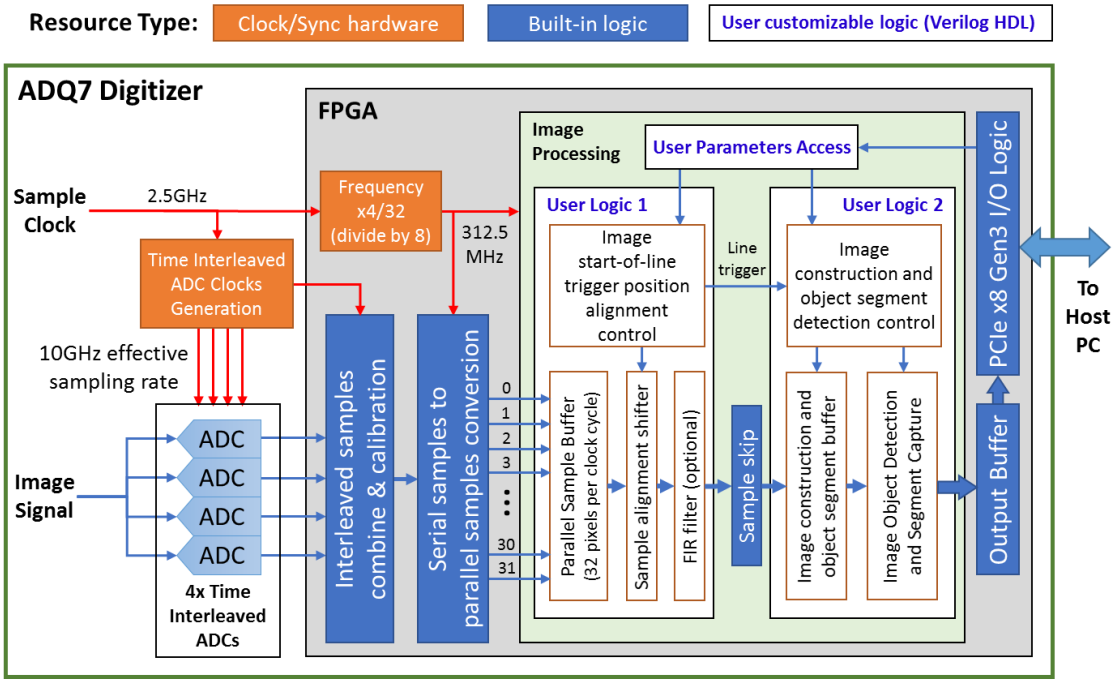

Figure S12 ADQ7 Digitizer hardware and FPGA programmable function layout

Implement the 2D image reconstruction and image processing/filtering (User Logic 2)

CRITICAL: The FPGA automatically generates a 2-bit digital signal (record_bits_in [1:0]) that dictates the initial start-of-line (SOL) and end-of-line (EOL) signals for use within User Logic 2 to perform the default 2D image reconstruction. It is generated according to a user defined trigger period and line length (record length) parameter configured through the API on the host PC (**Fig. S13**). The line length is defined in terms of number of pixels (K×32) where K is the number of data blocks (32 parallel pixels), and any K smaller than the trigger period will result in multiples of 32 pixels being dropped as shown in **Fig. S13**. User can modify the record bits and line length to arbitrarily crop the resulting 2D image in the direction of lines.

CRITICAL STEP: K must be less than or equal to the trigger period as illustrated in **Fig. S13**, and line length must be consistent to ensure correct 2D image construction.

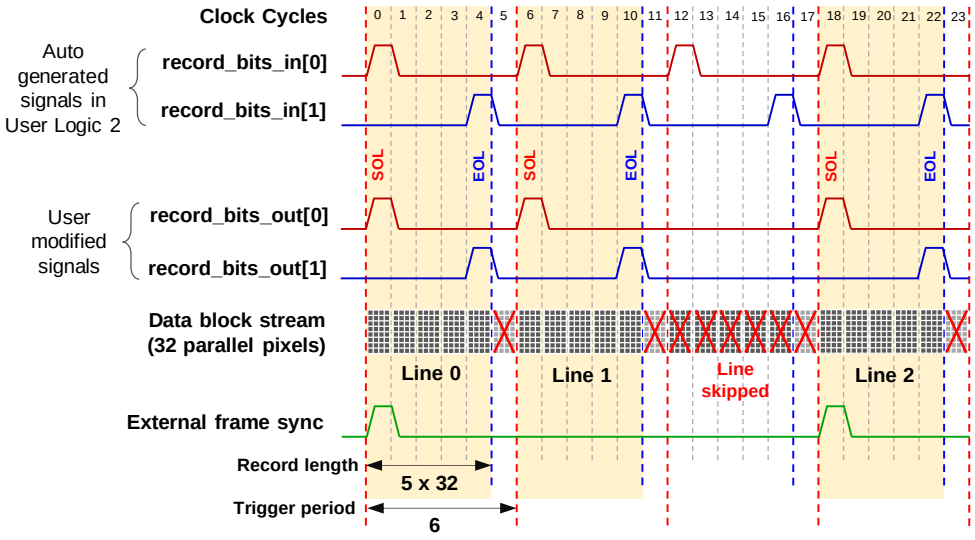

Figure S13 Line and frame synchronization signals in FPGA User Logic 2 module. SOL and EOL represent the start-of-line and end-of-line respectively. Note that all the synchronization signals are aligned to multiple of 32 pixels (1 data block). The record length represents the actual line length in number of pixels given by the number of clock cycles (K) multiplied by data block size (32). The actual line length is 5 × 32 = 160 pixels in this case.

1. Users can adaptively skip unwanted lines by removing SOL/EOL pairs in the initial record_bits_in[1:0] signals and return the resultant signals (record_bits_out[1:0]) as an output. This is particularly useful for adaptive data reduction, where unwanted pixel blocks, lines or 2D segments within the generated image stream can be dropped.
   CRITICAL STEP: User must take special care to ensure that both SOL and EOL are removed in pairs to prevent undesirable behaviors caused by unbalanced SOL/EOL signals.

2. External synchronization can also be applied through the TRIG or SYNC ports on the ADQ7 as a frame trigger (**Fig. S13**) to synchronize collection of lines (2D frame). The external frame trigger is particularly useful on system that utilizes galvo scanners where each scan corresponds to a 2D frame that must be synchronized with the FPGA for correct image frame alignment and processing.

3. Real-time segment/region-of-interest (ROI) detection from 2D image data stream:

   i. For the continuous 2D image streaming case where image frame size and frame trigger are not known in advanced, the FPGA needs to determine the lines that are of interest on-the-fly. The process is critical in reducing the data rate to within the PCIe channel's throughput, as well as the maximum throughput of the storage devices used, so that all data of interest can be recorded without dead time. For example, the ADQ7 PCIe operating at 10GS/s (2 bytes per sample) yields 20GB/s of data rate which is significantly greater than the throughput of ADQ7's PCIe x8 Gen3 interface. Taking data transfer overhead into account, the actual PCIe throughput is limited to approximately 6.6GB/s, so data reduction greater than a factor of 3 must be applied for sustained data transfer, assuming any storage device down the line can support such speed.

   ii. The algorithm used to identify lines of interest relies on detecting the gradient of pixel values between adjacent lines. The algorithm can be implemented in the User Logic 2 of the FPGA within ADQ7 using minimal amount of on-chip memory and basic arithmetic blocks. The main principle of the real-time region/segment detection hardware is described in **Fig. S14**. A reference design of the above image construction, filtering, data reduction and ROI detection can be found in Supplementary Software, under the folder "adq7_ul1_ul2_roi".

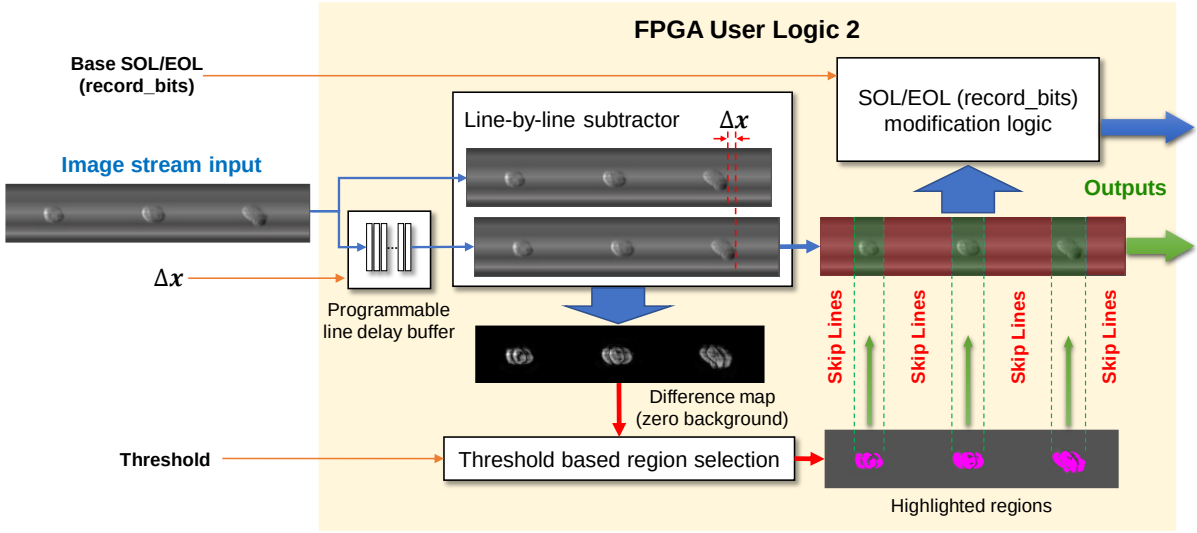

Figure S14 General schematic of the pixel-gradient-based cell region/segment detection hardware on FPGA for real-time data reduction. Any background patterns consistent along the horizontal direction are rejected in the process, and hence lines without cells/objects are identified and skipped.

Example case:

For an imaging system with line-scan rate of 1 MHz (1 µs period) and a galvo mirror scan rate of 1000 Hz (1000 fps), where each FACED line-scan has 80 foci with 2ns separation time between adjacent foci. Only 160ns (80 × 2ns) within each line is required to cover all fluorescence signals excited by the FACED foci. The remaining 840ns is dead time with “blank pixels”. Similarly, for each mirror scan (frame), there is a 100 µs dead time caused by mirror turns**.** Therefore, each frame contains 900 lines of useful data, followed by 100 “blank lines” during the dead time.

The ADQ7 can be configured as follows to perform continuous frame recording in the above case: Since the minimum sampling rate needed to cover the 80 foci per line-scan is significantly lower than the 10GHz sampling rate of the ADQ7, A sample-skip step size of 16 can be used to significantly lower the actual sampling rate to 625MS/s, achieving a 16x data throughput reduction. The built-in FIR filter can also be used prior to the sample-skip block to minimise any sample aliasing effect. The blank pixels in each line can be cropped by setting the line length to 128 pixels (4 blocks) via the host API, and then aligned by adjusting the record_bits in the FPGA to match the SOL with the start of the actual fluorescence signals. In the same way, each frame is synchronised by the galvo mirror reference trigger as an external TRIG signal to the FPGA, and blank lines at the end of each frame can be discarded through removing their corresponding SOL/EOL pairs as described earlier. The resultant throughput requirement is given by 1000fps × 900 lines × 128 pixels (16-bit) = 219.73MB/s, which is easily manageable by most medium to high-end storage devices within the host PC for long-term continuous image frame recording.

Supplementary Tables:

Table S1 List of all extracted biophysical phenotypes and the corresponding equations[1]

	Feature	Symbol	Equation
Bulk	Area*,**	A	$L_{pix}{}^2 \cdot N_{pix}$
	Volume**	V	$\frac{4}{3}\pi \cdot (\frac{L_{minor}}{2})^2 \cdot (\frac{L_{major}}{2})$
	Circularity*		$4\pi A/P$
	Eccentricity		$\frac{L_{ellip}}{L_{major}}$
	Aspect Ratio		$\frac{L_{minor}}{L_{major}}$
	Orientation		θ_{major}
Global	Peak Phase		$\max\{PG(x,y)\}$
	Phase Variance		$\frac{\iint_A (PG(x,y) - \overline{PG})^2 dxdy}{N_{pix} - 1}$
	Phase Skewness*		$\frac{\iint_A (PG(x,y) - \overline{PG})^3 dxdy / N_{pix}}{\sigma_{PGstd}{}^3}$
	Phase Kurtosis		$\frac{\iint_A (PG(x,y) - \overline{PG})^4 dxdy / N_{pix}}{\sigma_{PGstd}{}^4}$
	Phase Range**		$\max\{PG(x,y)\} - \min\{PG(x,y)\}$
	Phase Minimum*,**		$\min\{PG(x,y)\}$
	Phase Centroid Displacement**		$\sqrt{(x_{DMD,cen} - x_{cen})^2 + (y_{DMD,cen} - y_{cen})^2} \cdot L_{pix}$
	Mean Phase Arrangement		$\frac{\iint_A PG(r,\theta)\, r\, drd\theta}{\iint_A PG(r,\theta)\ drd\theta}$
	Phase Arrangement Variance	$\sigma_{PGarr}{}^2$	$\frac{\iint_A (PG(r,\theta)\, r)^2\ drd\theta}{\iint_A PG(r,\theta)\ drd\theta}$
	Phase Arrangement Skewness		$\frac{\iint_A (PG(r,\theta) \cdot r)^3\ drd\theta}{\sigma_{PGarr}{}^2 \cdot \iint_A PG(r,\theta)\ drd\theta}$
	Phase Orientation Variance	$\sigma_{PGang}{}^2$	$\frac{\int_0^\infty (\widetilde{PG}(\omega) \cdot \omega)^2\, d\omega}{\int_0^\infty \widetilde{PG}(\omega)\, d\omega}$
	Phase Orientation Skewness		$\frac{\int_0^\infty (\widetilde{PG}(\omega) \cdot \omega)^4\, d\omega}{\sigma_{PGang}{}^2 \cdot \int_0^\infty \widetilde{PG}(\omega)\, d\omega}$
Local	Phase STD Mean*	$\overline{PG_{STD}}$	$\frac{\iint_A PG_{STD}(x,y)\, dxdy}{N_{pix}}$
	Phase STD Variance*	$\sigma_{PGstd}{}^2$	$\frac{\iint_A (PG_{STD}(x,y) - \overline{PG_{STD}})^2 dxdy}{N_{pix} - 1}$
	Phase STD Skewness**		$\frac{\iint_A (PG_{STD}(x,y) - \overline{PG_{STD}})^3 dxdy / N_{pix}}{\sigma_{PGstd}{}^3}$
	Phase STD Kurtosis		$\frac{\iint_A (PG_{STD}(x,y) - \overline{PG_{STD}})^4 dxdy / N_{pix}}{\sigma_{PGstd}{}^4}$

	Phase STD Centroid Displacement		$\sqrt{(x_{PGSTD,cen} - x_{cen})^2 + (y_{PGSTD,cen} - y_{cen})^2} \cdot L_{pix}$
	Phase STD Radial Distribution*		$\frac{\iint_A r \cdot PG_{STD}(r,\theta)\, drd\theta}{\iint_A PG_{STD}(r,\theta)\, drd\theta}$
	Phase Entropy Mean*	PG_{ent}	$\frac{\iint_A PG_{ent}(x,y)\, dxdy}{N_{pix}}$
	Phase Entropy Variance	${\sigma_{PGent}}^2$	$\frac{\iint_A (PG_{ent}(x,y) - \overline{PG_{ent}})^2 dxdy}{N_{pix} - 1}$
	Phase Entropy Skewness		$\frac{\iint_A (PG_{ent}(x,y) - \overline{PG_{ent}})^3 dxdy / N_{pix}}{{\sigma_{PGent}}^3}$
	Phase Entropy Kurtosis		$\frac{\iint_A (PG_{ent}(x,y) - \overline{PG_{ent}})^4 dxdy / N_{pix}}{{\sigma_{PGent}}^4}$
	Phase Entropy Centroid Displacement*,**		$\sqrt{(x_{PGent,cen} - x_{cen})^2 + (y_{PGent,cen} - y_{cen})^2} \cdot L_{pix}$
	Phase Entropy Radial Distribution*		$\frac{\iint_A r \cdot PG_{ent}(r,\theta)\, drd\theta}{\iint_A PG_{ent}(r,\theta)\, drd\theta}$
	Phase Fiber Centroid Displacement		$\sqrt{(x_{PGfiber,cen} - x_{cen})^2 + (y_{PGfiber,cen} - y_{cen})^2} \cdot L_{pix}$
	Phase Fiber Radial Distribution**		$\frac{\iint_A r \cdot PG_{fiber}(r,\theta)\, drd\theta}{\iint_A PG_{fiber}(r,\theta)\, drd\theta}$
	Phase Fiber Pixel > Upper Percentile**		$\frac{Number\ of\ pixels\ in\ PG_{fiber}(x,y) > 75th\ percentile}{N_{pix}}$
	Phase Fiber Pixel > Median**		$\frac{Number\ of\ pixels\ in\ PG_{fiber}(x,y) > median}{N_{pix}}$

*Features used for the UMAP plot
**Features used for the PHATE 3D plot

Table S2 List of Variables Used in the Equations (Table S1):

Variable	Description	Equation
$\boldsymbol{C}$	Contour of binary mask	
$\boldsymbol{CM}$	Cell mask function	$CM(x,y) = \begin{cases} 1\ if\ inside\ cell \\ 0\quad otherwise \end{cases}$
$\boldsymbol{PGD}$	Phase gradient density map	$PGD(x,y) = \frac{\lambda \cdot PG(x,y)}{2\pi\alpha \cdot h(x,y)}$
$\boldsymbol{h}$	Cell height map	$h(x,y) = \sqrt{(\frac{L_{minor} + L_{major}}{2})^2 - ((x - x_{cen})^2 + (y - y_{cen})^2)}$
$\boldsymbol{L_{ellip}}$	Distance between foci of ellipse	
$\boldsymbol{L_{major}}$	Major axis length	
$\boldsymbol{L_{minor}}$	Minor axis length	
$\boldsymbol{L_{pix}}$	Physical length of one pixel	

$\boldsymbol{PG}$	Phase gradient map	$PG(x,y)$
$\boldsymbol{PG(\theta)}$	Phase gradient projected to polar angle	
$\boldsymbol{\widetilde{PG}(\omega)}$	Phase gradient in angular frequency domain	$\widetilde{PG}(\omega) = \mathcal{F}(PG(\theta))$
$\boldsymbol{\overline{PG_{STD,ker}}(x,y)}$	Mean value of PG image within STD filter kernel	$\frac{\int_{x-w_{STD}/2}^{x+w_{STD}/2}\int_{y-w_{STD}/2}^{y+w_{STD}/2} PG(u,v)\,dvdu}{{w_{STD}}^2}$
$\boldsymbol{PG_{STD}(x,y)}$	PG image STD map	$\int_{x-w_{STD}/2}^{x+w_{STD}/2}\int_{y-w_{STD}/2}^{y+w_{STD}/2}\sqrt{\frac{(PG(u,v)-\overline{PG}(x,y))^2}{{w_{STD}}^2}}\,dvdu$
$\boldsymbol{PG_{cubic}(x,y)}$	Cubic polynomial surface fit of phase gradient map	
$\boldsymbol{PG_{ent}(x,y)}$	Entropy filtered phase gradient map	$\sum_{k=0}^{255} p_{PG,k}\cdot\log_2 p_{PG,k}$
$\boldsymbol{PG_{fiber}(x,y)}$	Fiber texture enhanced phase gradient map	$FF(PG(x,y))$, (ref.[2])
$\boldsymbol{N_{pix}}$	Pixel number in cell mask	$\iint CM(x,y)\,dA$
$\boldsymbol{P}$	Perimeter	$\oint_C\sqrt{\left(\left(\frac{dx}{d\theta}\right)^2+\left(\frac{dy}{d\theta}\right)^2\right)}\,d\theta$
$\boldsymbol{p_{PG,k}(x,y)}$	Normalized histogram counts within kernel of phase gradient map	$p_{PG,k}(x,y) = \frac{number\ of\ pixels\ in\ kernel\ (w_{ent})\ with\ PG=k}{Total\ number\ of\ pixels\ in\ kernel}$
$\boldsymbol{r,\theta}$	Polar coordinates centered at cell centroid	
$\boldsymbol{w_{ent}}$	Kernel size of entropy filter	
$\boldsymbol{w_{STD}}$	Kernel size of STD filter	
$\boldsymbol{x,y}$	Cartesian coordinates	
$\boldsymbol{x_{cen}}$ $\boldsymbol{y_{cen}}$	Coordinates of cell centroid	$x_{cen} = \frac{\iint_A x\cdot CM(x,y)\,dxdy}{N_{pix}}$ $y_{cen} = \frac{\iint_A y\cdot CM(x,y)\,dxdy}{N_{pix}}$
$\boldsymbol{x_{PG,cen}}$ $\boldsymbol{y_{PG,cen}}$	Coordinates of PG weighted cell centroid	$x_{PG,cen} = \frac{\iint_A x\cdot PG(x,y)\,dxdy}{N_{pix}}$ $y_{PG,cen} = \frac{\iint_A y\cdot PG(x,y)\,dxdy}{N_{pix}}$
$\boldsymbol{x_{PGD,cen}}$ $\boldsymbol{y_{PGD,cen}}$	Coordinates of phase gradient density weighted cell centroid	$x_{PGDcen} = \frac{\iint_A x\cdot PGD(x,y)\,dxdy}{N_{pix}}$ $y_{PGDcen} = \frac{\iint_A y\cdot PGD(x,y)\,dxdy}{N_{pix}}$
$\boldsymbol{x_{PGent,cen}}$ $\boldsymbol{y_{PGent,cen}}$	Coordinates of entropy filtered PG weighted cell centroid	$x_{PGent,cen} = \frac{\iint_A x\cdot PG_{ent}(x,y)\,dxdy}{N_{pix}}$ $y_{PGent,cen} = \frac{\iint_A y\cdot PG_{ent}(x,y)\,dxdy}{N_{pix}}$

$x_{PGfiber,cen}$ $y_{PGfiber,cen}$	Coordinates of fiber enhanced PG weighted cell centroid	$x_{PGfiber,cen} = \frac{\iint_A x \cdot PG(x,y)\ dxdy}{N_{pix}}$ $y_{PGfiber,cen} = \frac{\iint_A y \cdot PG_{fiber}(x,y)\ dxdy}{N_{pix}}$
$x_{PGSTD,cen}$ $y_{PGSTD,cen}$	Coordinates of STD filtered PG weighted cell centroid	$x_{PGSTD,cen} = \frac{\iint_A x \cdot PG_{STD}(x,y)\ dxdy}{N_{pix}}$ $y_{PGSTD,cen} = \frac{\iint_A y \cdot PG_{STD}(x,y)\ dxdy}{N_{pix}}$
α	Specific refractive increment	$0.19\ ml/g$ (ref.[3])
θ_{major}	Angle between major axis and x-axis	
$\mathcal{F}$	Fourier transform	

1. Frangi, A. F., Niessen, W. J., Vincken, K. L. & Viergever, M. A. Multiscale vessel enhancement filtering. in *International conference on medical image computing and computer-assisted intervention* 130–137 (Springer, 1998).
2. Zhao, H., Brown, P. H. & Schuck, P. On the distribution of protein refractive index increments. *Biophys. J.* **100**, 2309–2317 (2011).

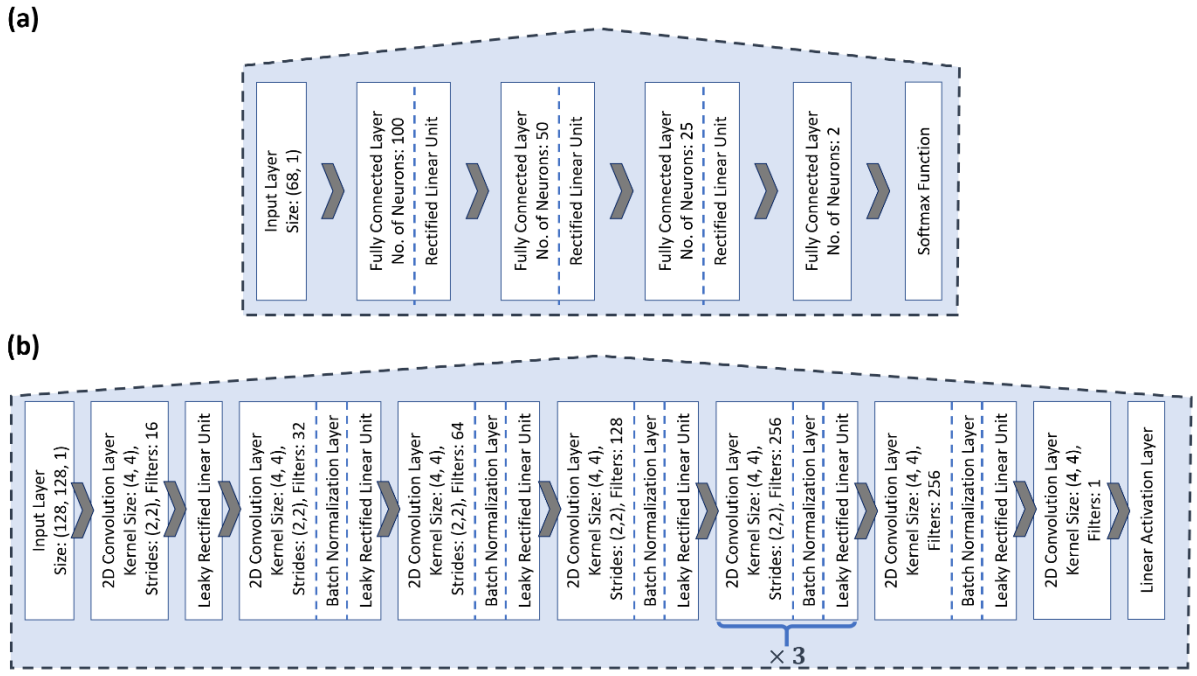

Figure S15 Architect of NN-based classification model and CNN-based regression model. (a) With the extracted cell features as the 1-dimensional input, the NN-based model was followed with 3 hidden fully connected layers, each with 100, 50 and 25 nodes respectively and linked with rectified linear unit as activation functions. To concrete the model output into the cell types classification, the model was ended with a fully connected layer with 2 neurons (number of classification classes). Softmax function was further employed in computing the probability of each cell type at the output layer and cross-entropy function was selected as loss function. **(b)** The CNN-based regression model was composed of multiple 2D convolution layers, batch normalization layers and leaky rectified linear units. At the output layer, linear activation function was used to predict the normalized fluorescent intensity of cells, which was indicative on the cell cycle status.

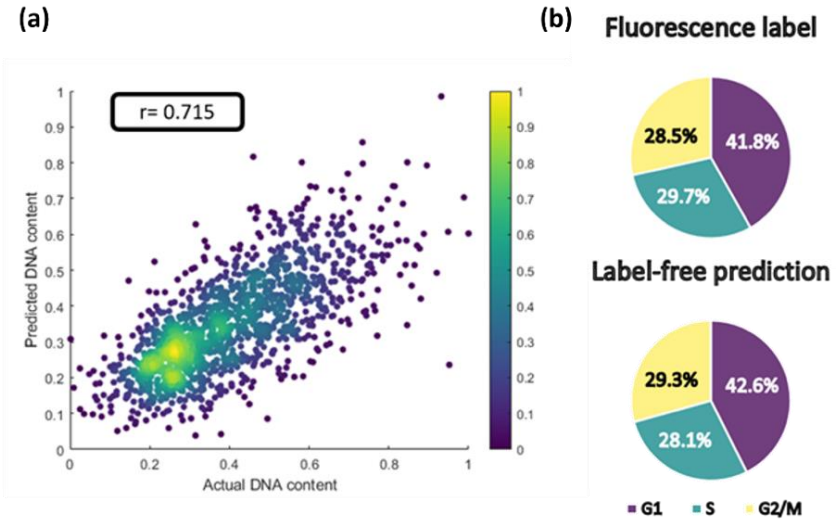

Figure S16 Results of predicting the DNA content. (a) Evaluation of the correlation between the predicted DNA content and the actual DNA content. **(b)** Comparison between the predicted proportions of G1, S and G2 phases with the ground truth based on the fluorescence label.

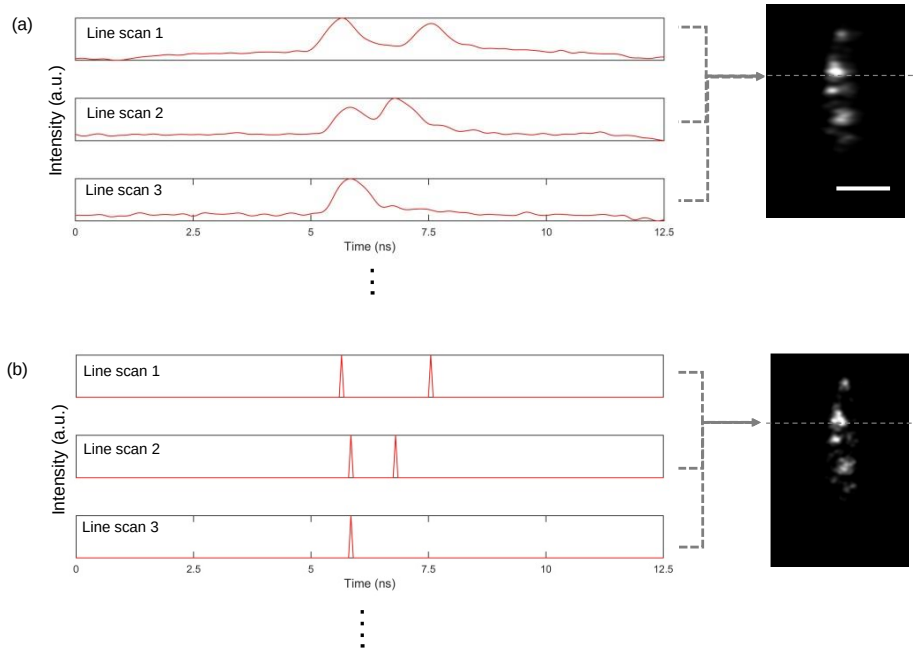

Figure S17 Time-gated photon counting detection in FACED SHG imaging. (a) SHG line-scan capture without time-gated photon counting detection. (b) The same SHG line-scan capture with time-gated detection. The sampled time window is highlighted as yellow shaded area. Scale bar:10μm.