Methods and Applications in Fluorescence

PAPER

CrossMark

Compact and robust deep learning architecture for fluorescence lifetime imaging and FPGA implementation

Zhenya Zang[1] , Dong Xiao[1], Quan Wang[1], Ziao Jiao[1], Yu Chen[2] and David Day Uei Li[1]

1   Department of Biomedical Engineering, University of Strathclyde, Glasgow G1 1XQ, United Kingdom
2   Department of Physics, University of Strathclyde, Glasgow G4 0NG, United Kingdom

E-mail: david.li@strath.ac.uk

Abstract

This paper reports a bespoke adder-based deep learning network for time-domain fluorescence lifetime imaging (FLIM). By leveraging the l_1-norm extraction method, we propose a 1D Fluorescence Lifetime AdderNet (FLAN) without multiplication-based convolutions to reduce the computational complexity. Further, we compressed fluorescence decays in temporal dimension using a log-scale merging technique to discard redundant temporal information derived as log-scaling FLAN (FLAN +LS). FLAN+LS achieves 0.11 and 0.23 compression ratios compared with FLAN and a conventional 1D convolutional neural network (1D CNN) while maintaining high accuracy in retrieving lifetimes. We extensively evaluated FLAN and FLAN+LS using synthetic and real data. A traditional fitting method and other non-fitting, high-accuracy algorithms were compared with our networks for synthetic data. Our networks attained a minor reconstruction error in different photon-count scenarios. For real data, we used fluorescent beads' data acquired by a confocal microscope to validate the effectiveness of real fluorophores, and our networks can differentiate beads with different lifetimes. Additionally, we implemented the network architecture on a field-programmable gate array (FPGA) with a post-quantization technique to shorten the bit-width, thereby improving computing efficiency. FLAN+LS on hardware achieves the highest computing efficiency compared to 1D CNN and FLAN. We also discussed the applicability of our network and hardware architecture for other time-resolved biomedical applications using photon-efficient, time-resolved sensors.

1. Introduction

Fluorescence is represented by detectable photon emission from molecules or atoms, a radiative process transferring energy from an excited state to a lower state. Fluorescence lifetime is a quantitative metric to analyze fluorescence phenomena. Fluorescence lifetime imaging (FLIM) is an emerging technique boosting biomedical imaging. For example, endogenous fluorophores, such as amino acids, metabolic enzymes, and vitamins, have been engineered to assess *in Vivo* imaging [1, 2]; exogenous fluorophores, such as Förster resonance energy transfer (FRET) pairs, have been adopted to investigate protein interactions [3] and drug developments [4]. Autofluorescence has been utilized in fluorescence-guided surgery [5, 6]. Time-correlated single-photon counting (TCSPC) [7] is a powerful technique for time-resolved, wide-field

[8], and confocal FLIM [9] as it exhibits high signal-to-noise ratios (SNR) of limited photons emitted from fluorophores [10]. To guarantee precise time-of-arrival, a stopwatch employing high temporal resolution (picosecond) time-to-analogue or -digital converters can accumulate arrival photons from a train of laser pulses and generate a histogram decay. Retrieving lifetimes from temporal point spread functions (TPSF, or histogram decays of collected photons) is an ill-posed problem in which a perfect solution cannot be found. Traditional deconvolution-based nonlinear square fitting (NLSF) [11] and maximum likelihood estimation [12] methods are computationally expensive.

Fast, model-free methods such as centre-of-mass [13] and its derivatives [14] were proposed to estimate lifetimes without iterations to overcome the mentioned issues. However, they are susceptible to low

SNRs. They can resolve either mono- or bi-exponential decays, meaning it is hard to analyze complex fluorophores with multi-exponential decays. Moreover, they cannot simultaneously and accurately recover two critical average lifetimes, i.e., amplitude- (τ_A) and intensity-averaged lifetime (τ_I). The selection of the two average lifetimes should be considered for different applications [15]. This study clarified that τ_A is a crucial parameter to deduce energy transfer efficiency and dynamic quenching behaviours; τ_I is essential to solving the quenching constant from the Stern-Volmer constant. The study [16] clarified that the ratio of τ_A to τ_I is an intuitive indicator for monitoring multi-exponential fluorescence decays. Therefore, we aim to construct a model-free and robust method to reconstruct τ_A and τ_I to cover most FLIM applications. Given that generating artificial FLIM TPSFs is well-documented, data-driven approaches can boost the accuracy and computing efficiency of FLIM analysis. Therefore, we constructed a compact hardware-friendly deep learning (DL) architecture for high-accuracy τ_A and τ_I reconstruction. Further, we introduced a nonlinear mapping approach to compress histograms without performance degradation.

On the other hand, an increasing number of computing platforms for biomedical devices are migrating toward the edges. For example, portable oximeters can measure oxygen concentration of flowing blood non-invasively using photoplethysmography. Such highly integrated devices perform efficient computing, yet the power budget is considerably low. Likely, we yearn to implement our DL networks on FPGA as it can achieve highly parallel, reconfigurable computing while consuming low power. We designed an automatic software script to extract, and export learned parameters from pre-trained models to FPGA, paving the way for other time-resolved biomedical applications. Further, we adopted on-chip post quantization to shorten the bit-width of the on-the-fly data. We analyzed the computing efficiency of the FPGA embedded with our neural networks and compared it with CPU and GPU.

The following structure of the paper is that section 2 reviews and summarizes the existing FLIM algorithms on hardware; section 3 mathematically clarifies the optical characterization of FLIM and explains lifetime parameters to be resolved; section 4 demonstrates the network architecture and training details; section 5 describes the data compression strategies; section 6 and 7 evaluate our networks using synthetic and real data, respectively; section 8 reveals details of networks' FPGA implementation; section 9 conclude this work and discuss the potential improvements.

2. Prior work

2.1. Hardware platforms for FLIM
Graphics processing units (GPUs) have been extensively adopted as hardware accelerators for FLIM.

Byungyeon *et al* [17] proposed a mean-delay technique and implemented it on a GPU to directly extract fluorescence lifetime from analogue fluorescent decays. Additionally, a GPU was utilized to realize a pixel-wise, fitting-free phasor method [18] towards a two-photon FLIM system to achieve video-rate fluorescence lifetime inference. A compressed-sensing algorithm (CS) was applied to a high-resolution, wide-field FLIM system that used compressed ultrafast photography [19]. A GPU-based computer cluster was used to solve the iterative reconstruction algorithm. An FPGA is the ideal hardware for data readout and computing due to its sufficient I/O interfaces and reconfigurable computing parallelism. Particularly, single-photon avalanche diode (SPAD) arrays [20] coupled with FPGA as parallel and high-throughput data readout are needed. Therefore, some FPGA-friendly FLIM algorithms were proposed. Embedded center-of-mass (CMM) algorithms were implemented on FPGAs [21] and application-specific integrated circuits (ASICs) [22, 23] to determine lifetimes from time-to-digital converters' time stamps directly. Another CMM processor was implemented on an FPGA coupled with a SPAD-based silicon photomultiplier sensor [24], realizing real-time fluorescence lifetime estimation and online fluorescence lifetime cytometric sorting. After that, a DL prototype that is less susceptible to noise [25] was embedded on an FPGA to estimate fluorescence lifetime in a fluorescent flow cytometry system. On-chip lifetime estimation was also realized in the frequency domain; discrete Fourier transform was embedded on an FPGA to compute phase lifetimes and modulations [26].

2.2. Deep learning for FLIM
In the recent decade, DL has been an emerging approach for FLIM analysis. An MLP [27] was initially reported to analyze bi-exponential decays for a two-photon system. To build a noise-robust DL architecture, a 1D convolutional neural network (CNN) [28] was proposed to resolve bi- and tri-exponential decays with low SNRs. An MLP [29] was applied on a high spatial resolution, wide-field FLIM using time-gated SPAD without raster-scanning to miniaturize the FLIM system. Another MLP [30] processes phasor coordinates and generates lifetime parameters. Because the inputs are coordinates rather than histograms, the MLP exhibits fast speed. To extract both spatial and temporal features from raw fluorescence data tensors, a 3D CNN [31] was reported for near-infrared FLIM applications. Besides, Ruoyang *et al* proposed a 3D CNN combined with CS [32] for wide-field FLIM imaging *in vitro* and *in vivo* environments. Further, to address the challenge of FLIM with photon-starved conditions, Yuan-I *et al* reported a generative adversarial network [33] for FLIM imaging in low-photon scenarios (below 400). DL was also adopted to unmix fluorescence coefficients from

spectrally overlapped fluorophores. Jason *et al* proposed a hybrid CNN architecture [34] to extract individual lifetime components from multi-exponential, hyperspectral fluorescence emission decays. To enhance the spatial resolution of FLIM images, Dong *et al* [35] introduced a cascade CNN architecture to infer fluorescence lifetime and improve the spatial resolution afterwards. An Extreme Learning Machine (ELM) [36] was presented to achieve fast and accurate lifetimes reconstruction with back-propagation-free, online training.

3. Problem definition

Our initial objective is to retrieve lifetime parameters from TPSFs, where TPSF $h(t)$ for each focal point can be formulated by the convolution of the periodic IRF $I(t)$ and the multi-exponential probability density functions (PDF) $p(t)$

$$h(t) = I(t) * p(t) + \varepsilon(t)$$
$$= I(t) * \left[A \sum_{i=1}^{N} a_i \exp(-t/\tau_i) \right] + \varepsilon(t), \quad (1)$$

where $\sum_{i=1}^{N} a_1 = 1$, A depicts the amplitude, a_i and τ_i means the amplitude fractions and lifetime of an individual lifetime component, N implies the number of lifetime components. As Poisson noise is inevitable in TCSPC systems, $\varepsilon(t)$ represents Poisson noise which is the square root of the total photons in each time bin in a decay curve. The intensity-averaged lifetime τ_I means the average lifetime a fluorophore stays excited, depicted by

$$\tau_I = \frac{\int_0^\infty t \sum a_i \exp(-t/\tau_i) dt}{\int_0^\infty \sum a_i \exp(-t/\tau_i) dt} = \frac{\sum_{i=1}^{N} a_i \tau_i^2}{\sum_{i=1}^{N} a_i \tau_i}. \quad (2)$$

An amplitude-averaged lifetime is the integral of an intensity decay that is equivalent to steady-state intensity, formulated by

$$\tau_A = \int_0^\infty \sum a_i \exp(-t/\tau_i) dt = \sum_{i=1}^{N} a_i \tau_i. \quad (3)$$
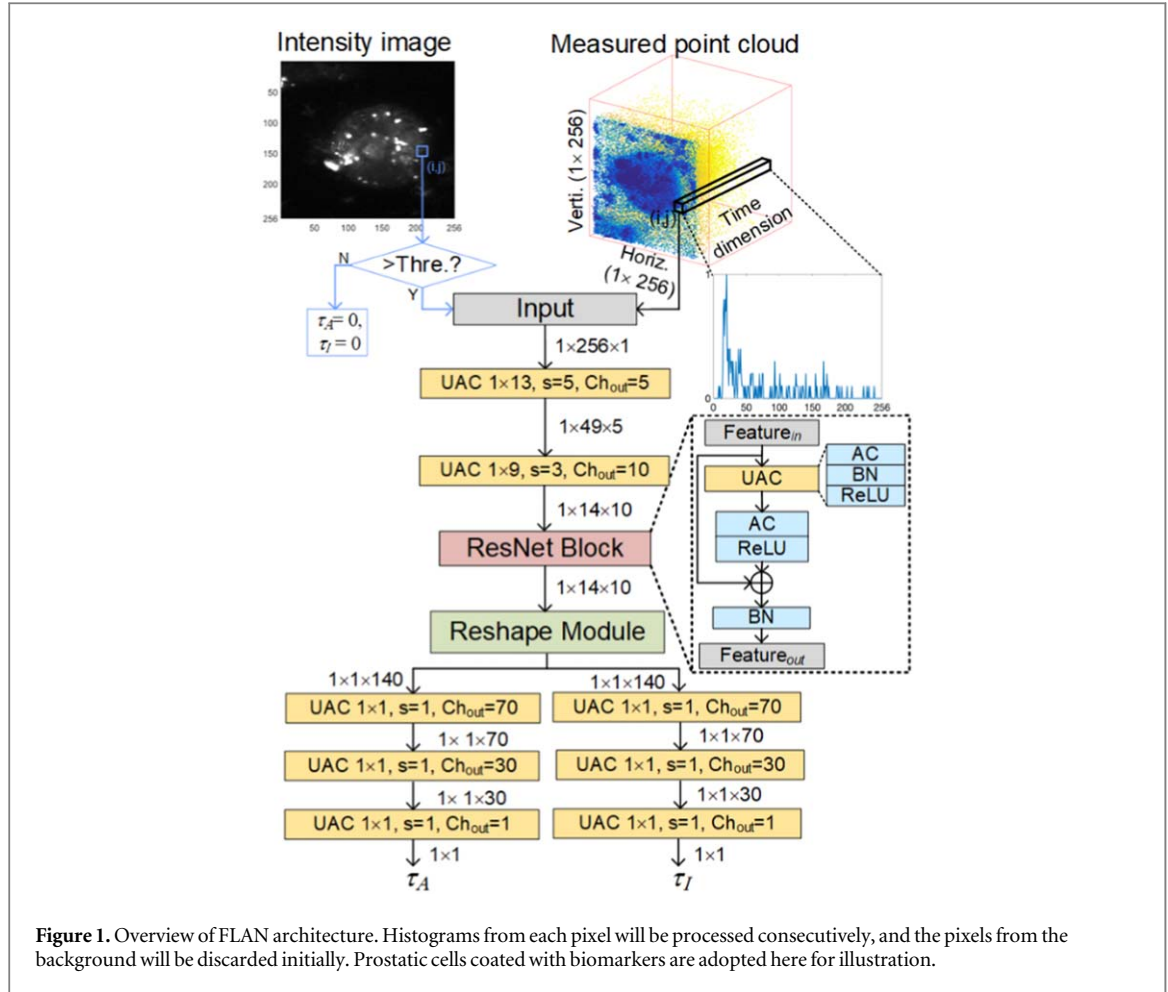
4. Deep learning network details

We intend to achieve high-fidelity FLIM images with a minor computing overhead. The extreme case of lightweight DL networks is the binarized-neural network [37] that only involves bit-wise operations. However, it suffers from low accuracy in practical applications [38]. It was proved that 2D AdderNet [39] could achieve comparable accuracy with CNN without matrix multiplications. Here, we introduce a derivative of the original AdderNet, compressed 1D fluorescence lifetime AdderNet (FLAN), catering to the histogram decays' processing.

4.1. Network architecture

The naïve AdderNet was initially used for 2D image classification [39]. It has been proved that adder-convolutions (AC) using l_1 distance to calculate similarities between feature maps (FM) and kernels can achieve nearly identical accuracy versus traditional convolutions. We use strides in each AC instead of zero-padding to simplify hardware logic to enlarge reception fields. The schematic of FLAN is shown in figure 1. We constructed unified AC (UAC) modules including AC, ReLU and batch-normalization (BN) to facilitate hardware implementation. And a ResNet [40] block was placed in the network's backbone to alleviate gradient vanishing. Differentiating from our previous work [28] that extracting individual lifetime components, we adopted a branched topology to infer two average lifetimes τ_A and τ_I that can analyze most FLIM applications. Algorithm 1 demonstrates the entry and the first layer of FLAN. A pre-defined threshold (T) was applied to filter out background pixels (Line 1). If the pixel is not from the background ($N_{pc} > T$), a UAC will process the histogram (from Line 2 to 11). Moreover, we proposed FLAN + LS that compresses histograms using log-scale nonlinear mapping. FLAN + LS needs fewer down-sampling layers because the length of input histograms was compressed in advance. Therefore, FLAN + LS is more compact and saves less computing overhead than FLAN. The details of compression will be illustrated in section 5.

We compared the size of existing DLs for FLIM, and the compression ratio is defined by the ratio of FLAN + LS's parameter size to uncompressed models. Table 1 summarizes current DL architectures for FLIM. Among the existing architectures that can resolve multi-exponential decays, our models obtain the smallest parameter size apart from Phasor-MLP [39]. Phasor-MLP has the fewest parameters as histograms were initially converted into phasor coordinates using the Fourier transform. Despite the simplicity, reconstructed lifetime parameters sometimes had distinct dispersion than 3D CNN [31]. ELM [36] has the simplest structure containing only one hidden layer.

In contrast, the hidden layer contains 500 nodes. ELM's computational complexity exponentially increases as the number of nodes grows. Such a complex, fully connected structure is not efficient for hardware implementation. Besides, although the 1D CNN [25] for fluorescence flow cytometry has a compact size and was optimized and implemented on FPGA, it cannot analyze multi-exponential decays. As the data throughput of PMT or SPAD array is enormous (Gigabyte per second), lightweight DL architecture and efficient data compression are imperative to alleviate the bandwidth of the data transfer. FLAN and FLAN + LS can efficiently address the issues with the ACs and histogram compression method.

Figure 1. Overview of FLAN architecture. Histograms from each pixel will be processed consecutively, and the pixels from the background will be discarded initially. Prostatic cells coated with biomarkers are adopted here for illustration.

4.2. Preparation for hybrid training data

As bi-exponential models could well approximate multi-exponential decays [16]. We generated hybrid datasets containing mono- and bi-exponential decays according to (1). The generation flow is shown in figure 2. The synthetic IRF with 0.167 ns Full width at half maximum (FWHM) was simulated following a Gaussian curve centre at the 14th time bin that is consistent with our two-photon system, given by

$$I(t) = \exp\left\{-2\left[\frac{(t-14)\,2(0.5\ln 2)^{1/2N}}{FWHM}\right]^{2N}\right\},$$

(4)

where we set $N = 1$ to make the synthetic IRF close to the real IRF. The mono- and bi-exponential PDFs were generated using the same instrumental parameters. Then bespoke TPSF were generated by convolving the IRF with PDFs. Poisson noise was added, ultimately using *Poissrnd* () in MATLAB to mimic background and shot noise. Peak photon counts (N_p) and lifetime parameters (τ for mono-exponential, τ_1, τ_2, and α for bi-exponential decays) were randomly selected in given ranges, as shown in table 2. Each synthetic TPSF was labelled with GT lifetime parameters to train our networks afterwards. Then, these two types of TPSFs were shuffled to finalize the datasets.

4.3. Training details

FLAN and FLAN + LS architectures were implemented by PyTorch and trained on a 16 GB NVIDIA Quadro RTX 5000 GPU. To accelerate the training, we employed adaptive learning rates (ALR) with an initial value of 0.001 and a momentum factor of 0.995. RMSProp is the optimizer in the training phase. 50,000 synthetic decays were generated for training purposes; extra 5,000 decays were utilized for validation. Early-stopping with 20-epoch patience was adopted to prevent overfitting. ReLU is the activation function after each AC layer. BN was used to mitigate the internal covariate shite of FMs in the network, whereby the training can be accelerated. The adopted loss function is MSE, depicted as

$$\mathcal{L}(\Theta) = \frac{1}{B_1}\sum_{b_1}^{B_1}\sum_{b_2}^{B_2} \left\| F_{b2}(h^{b_1}, \Theta) - G^{b_1} \right\|_2^2,$$

(5)

where B_1 and B_2 are the batch size (128) and the number of output nodes (2), $F(\cdot)$ is the end-to-end inference function of training parameters Θ, and G is the set of ground truth lifetimes. To make comparisons, we re-trained the 1D CNN [28] with traditional multiplications-based convolutional kernels for comparison, where we kept the networks' topology identical to FLAN and FLAN + LS. As the learning rate (0.0001) of the original 1D CNN is constant, it would

Algorithm 1. Unified AC for each histogram.

Input: input FM F_i of shape $CH_i \times W_i$;
    filters W of shape $K_x \times CH_i \times CH_o$;
    BN coefficients *scale* and *shift* of shape CH_o;
    stride S; Threshold T; photon count N_{pc};
Output: output FM F_o of shape $CH_o \times W_o$;
1:   **if** $N_{pc} > T$ **then**
2:   **for** $ch_i = 0;\ ch_i < CH_i;\ ch_i{++}$ **do**
3:    **for** $w_o = 0;\ w_o < W_o;\ w_o{++}$ **do**
4:     **for** $k_x = 0;\ k_x < K_x;\ k_x{++}$ **do**
5:      $w = ch_i \times S + k_x$;
6:      **if** $(w >= 0)\&\&(w <= W_i)$ **then**
7:       **while** $ch_o < CH_o$ **do**
8:        $F_o[w_o][ch_o] += -|F_i[w][ch_i]$
         $-W[k_x][ch_i][ch_o]|$;
9:        $ch_o{++}$;
10:   $F_o = F_o \times scale[ch_o] + shift[ch_o]$;
11:   $F_o = ReLU(F_o)$;
12: **else**
13:   $F_o = 0$;

---

exhibit a slower convergent speed than ALR. The loss curves of the three networks are shown in figure 3. FLAN + LS achieves a similar loss curve versus FLAN even with fewer time bins. And these two networks obtained similar ending loss versus 1D CNN when training ceased.

5. Data compression strategies

We introduce two-fold data compression methods to alleviate computing latency and reduce data movements. At first, we compressed histograms using a log-scale bin-merging technique to reduce redundant time bins. And then, a post-linear-quantization was employed on FPGA to compress bit-width throughout the FLAN and FLAN + LS architecture.

5.1. Log-scale mapping for time-bin

The fluorescence decays are less informative as the index of time-bin increases, especially for the specimen with ultra-small lifetimes. As blue lines show in figure 4, they are mono-and bi-exponential decays with lifetimes of 0.5 ns and 0.45 ns, respectively. No photon occurred after 86th and 84th time-bin, meaning that time bins behind them can be merged to compress the histogram and moderate the sparsity. Therefore, the computational complexity can be decreased. The target number of time bins M was controlled by a factor that should be predefined; the time-bin interval of the compressed histogram is

$$S = s(x+1) - s(x) =$$
$$floor\left(\frac{r^x - 1}{r-1}\right) - floor\left(\frac{r^{x+1} - 1}{r-1}\right), \quad (6)$$

where *floor* $(\cdot)$ rounds each element to the nearest integer less than or equal to that element, T is the original number of time-bin (256), x is the index of compressed time-bin. r is calculated from M, given by

$$\frac{r^M - 1}{r-1} = T, \quad (7)$$

where M is 80 in our experiments because the compressed histograms can reserve sufficient temporal information after merging. The number of photons in the new time interval is the summation of the counts from the original intervals. The nonlinear mapping process is depicted in figure 5; more temporal information can be reserved for the front bins. Accordingly, in figure 4, both typically synthesized mono- and bi-exponential decays with 256 time-bin can be effectively converted into 80 bins. Compressed decays with fewer time-bin will simplify the neural network. For the decays with greater lifetimes, M can be easily adjusted; and the training data can be generated accordingly.

5.2. On-chip linear quantization

Quantization for FMs and learned parameters are well-known approximate arithmetic for DL. And it is economical for logic utilization and power consumption on FPGA. We adopted the linear quantization (or static fixed-point) strategy [41] to convert floating-point (FLP) learned parameters and FMs in pre-trained models into fixed-point (FP). To maintain the computing accuracy, we configured FMs with 16-bit integer and 16-bit fractional length segmentation; learned parameters with 10-bit integer and 10-bit fractional parts. The conversion can be depicted as

Table 1. Comparisons of deep learning architectures for FLIM.

	3D CNN [31]	1D CNN [25]	1D CNN [28]	MLP [27]	Phasor-MLP [30]	MLP [29]	GAN [33]	ELM [36]	**FLAN**	**FLAN + LS**
Multi-exponential compatible	Yes	No	Yes	Yes	Yes	No	Yes	Yes	Yes	Yes
Input data type	3D tensor	1D histogram	1D histogram	1D histogram	Phasor coordinates	1D histogram	1D histogram	1D histogram	1D histogram	1D histogram
Model size	4.14MB (1,084,045)	0.016MB (4,248)	0.19MB (48,675)	0.57MB (149,252)	1.84KB (471)	14.31MB (3,750,250)	0.55MbB (143,528)[a]	0.13MB (205,600)	0.088MB (23,003)	0.017 MB (4,058)
Training time	4 h	17 min	23 min	4 h	15 min	38 min	6.9 h	10.85 sec	20 min	18 min
Training hardware	GPU	GPU	GPU	GPU	CPU	GPU	GPU	CPU	GPU	GPU
Inferencing hardware	GPU	GPU or FPGA	GPU	GPU	CPU	GPU	GPU	CPU	GPU or FPGA	GPU or FPGA
Data compression	—	Quantization-aware training	—	—	—	—	—	—	Post-quantization	Post-quantization+LS

[a] The model size was not given in the paper; the size was calculated from the network's structure.
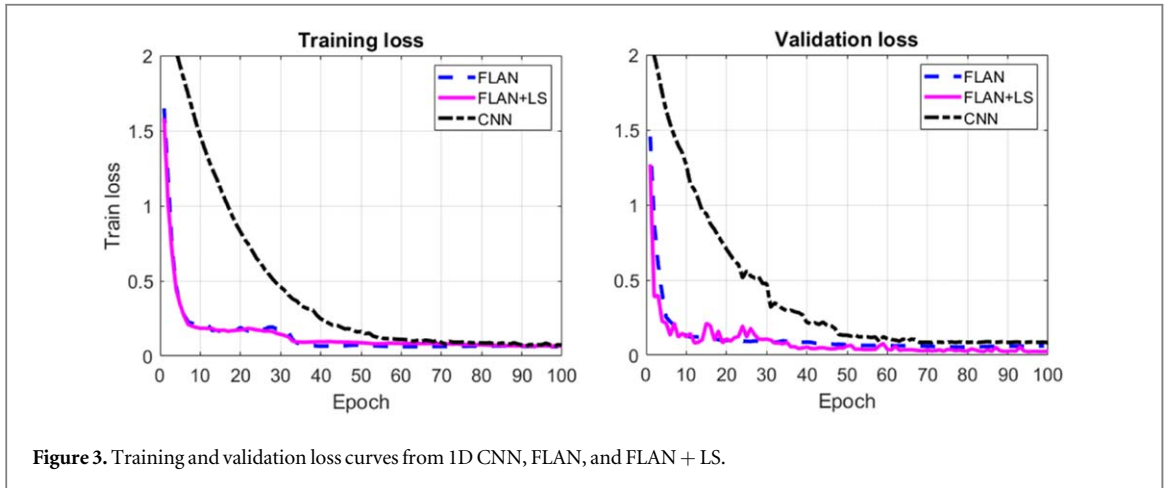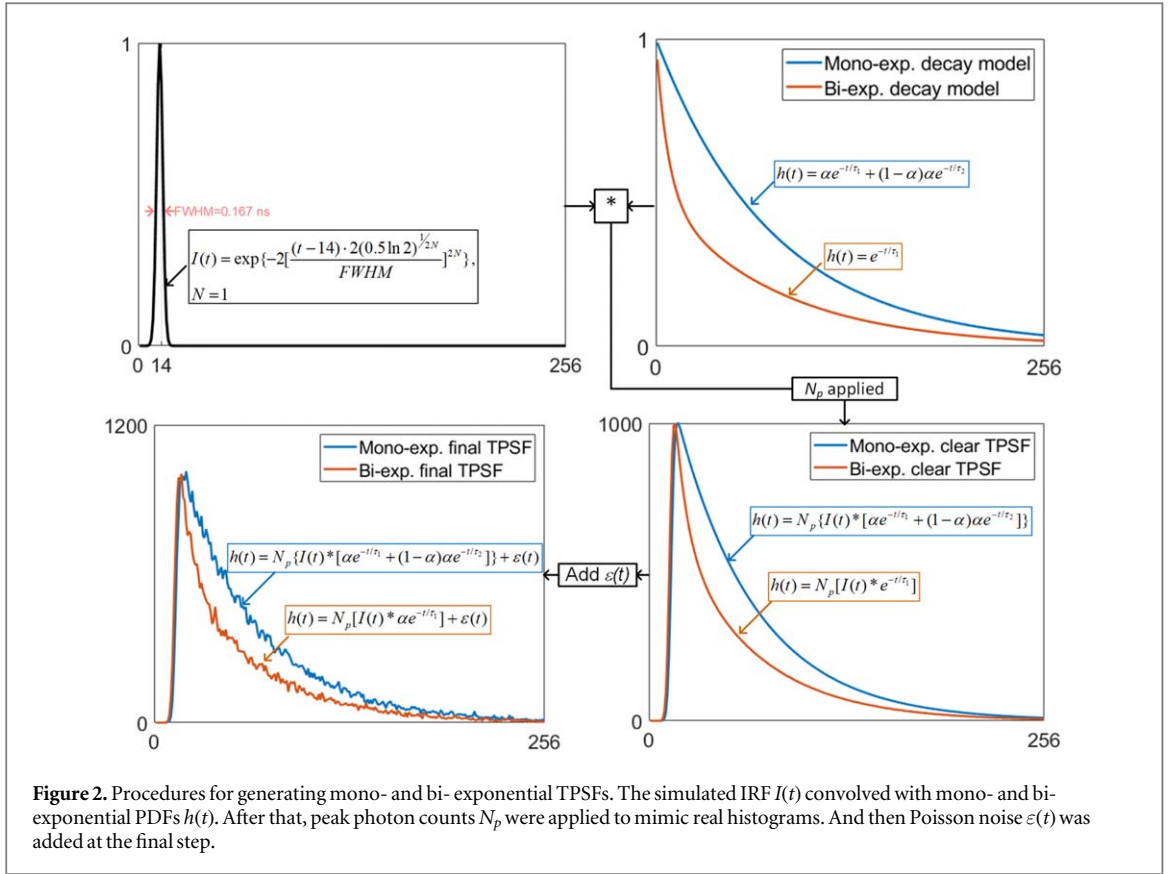
Figure 2. Procedures for generating mono- and bi- exponential TPSFs. The simulated IRF $I(t)$ convolved with mono- and bi-exponential PDFs $h(t)$. After that, peak photon counts N_p were applied to mimic real histograms. And then Poisson noise $\varepsilon(t)$ was added at the final step.

Figure 3. Training and validation loss curves from 1D CNN, FLAN, and FLAN + LS.

Table 2. Ranges of lifetime parameters for synthetic data.

	τ_1 (ns)	τ_2 (ns)	a	τ_I (ns)	τ_A (ns)	N_p
Mono-exp	[0.1, 5]	—	1	[0.1, 5]	[0.1, 5]	[10, 400]
Bi-exp.	[0.1, 0.5]	[1, 3]	[0, 1]	[0.01, 3.98]	[0.01, 3.98]	[10, 400]

$$\begin{cases} D = round(N_{FLP} \times 2^F) \\ B = D2B(B) \\ N_{FL} = shift(B, F) \end{cases}, \qquad (8)$$

where F is the fractional length of the FP number, $D2B(\cdot)$ means converting from decimal to binary type, $shift(\cdot)$ moves the decimal three places to the left, and N_{FLP} and N_{FP} depict the FLP number and binarized FP number, respectively. The precision of the FP number is 2^{-10}. The quantization can be realized using FP libraries in Vivado High-Level-Synthesis (HLS).
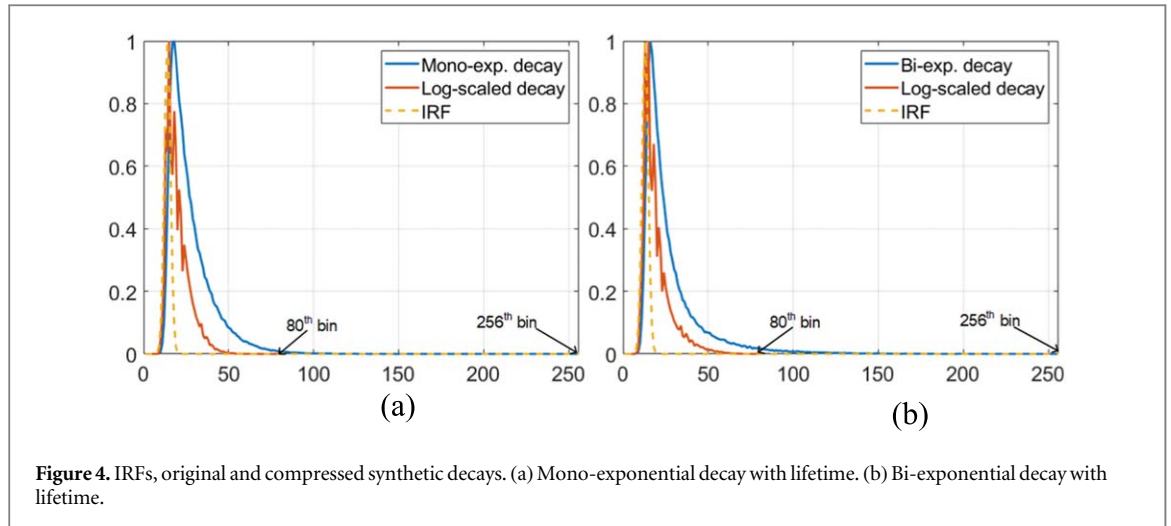
Figure 4. IRFs, original and compressed synthetic decays. (a) Mono-exponential decay with lifetime. (b) Bi-exponential decay with lifetime.

6. Synthetic data evaluation

With hybrid test datasets containing shuffled mono- and bi-exponential decays, we quantitatively evaluated FLAN and FLAN + LS in terms of τ_A and τ_I reconstruction. And we compared them with traditional curve fitting and other high-accuracy algorithms in different photon-count conditions. To generate intuitive 2D GT FLIM images, two individual lifetime components (τ_1 and τ_2 in (1)) were fixed values of 0.3 ns and 2.5 ns. And from the top to the bottom of GT lifetimes, amplitude fractions uniformly increased from 0 to 1. Mean square error (MSE) was used to evaluate the accuracy of each method in figure 6. Three deep learning methods achieved better recovery performance for both τ_A and τ_I. And both FLAN and FLAN + LS outperformed 1D CNN under different photon-count conditions with the same training datasets. Despite the compression of FLAN + LS, the performance was not deteriorated, even in low-count conditions. In terms of τ_A, the performance of NLSF declined as photon counts decreased because the fitting process (Levenberg-Marquart deconvolution) required enough counts to guarantee accuracy. The provocative study [42] using similar synthetic parameters to ours unraveled that, for mono-exponential decays, the necessary number of photons of the deconvolution method is around 100 for the minimal resolvable lifetime of 0.3 ns. Moreover, pre-set initial values are critical for the deconvolution method, and we did not fine-tune them here. BCMM was also susceptible to the low-count condition (below 100) as Romberg's integration in BCMM is inaccurate for sparse vectors. As for τ_I, in figures 6(d) and (e)., although CMM achieved smaller MSE than deep learning methods under high and medium photon counts conditions, discernable bias can be observed in the top half of CMM's recovered images. Before using CMM, prior knowledge about the width of the analysis window is required to achieve optimal results. To observe the lifetime distributions more

efficiently, we visualized the lifetime distributions of all recovered and GT images in figure 7.

7. Real-case study: fluorescent beads discrimination

To evaluate our networks using fluorescent samples, we used different algorithms to retrieve lifetimes based on fluorescent beads with characterized lifetimes. Moreover, we used convallaria majalis cells with short acquisition time (3 seconds) to validate the accuracy of our algorithms in real low counts conditions. This section introduces the sample preparation processes, optical setup, and evaluation results.

7.1. Sample preparation

We chose fluorescent latex beads to evaluate the accuracy of our networks as they are intuitive tools applied to tracing fluid dynamics [43] and validating hyperspectral imaging microscopy [44]. Here, yellow-green and crimson fluorescent beads of 10 µm diameter (F8831, FluoSphere Polystyrene Microspheres, Thermo Fisher, UK) were dissolved in an aqueous concentration of 3.6×10^5 beads/ml. We dropped the diluted solution of fluorescent beads onto a microscope slide covered with coverslips and then mounted it on microscope sample holder of a two-photon system. The unmixed set encompasses yellow-green beads with a reference lifetime of 2.1 ns. Alone with the unmixed group, the mixed set includes extra crimson beads with a 3 ns reference lifetime. The reference lifetime was measured using the TCSPC technique on a Horiba Deltaflex fluorometer and analyzed using Horiba DAS6 software.

7.2. Optical setup

The utilized excitation source was a femtosecond Ti: sapphire laser (Chameleon, Coherent, Santa Clara, USA) with tuneable wavelengths, and the laser pulse width is less than 200 fs with an 80 MHz repetition rate. The emission light was collected by a
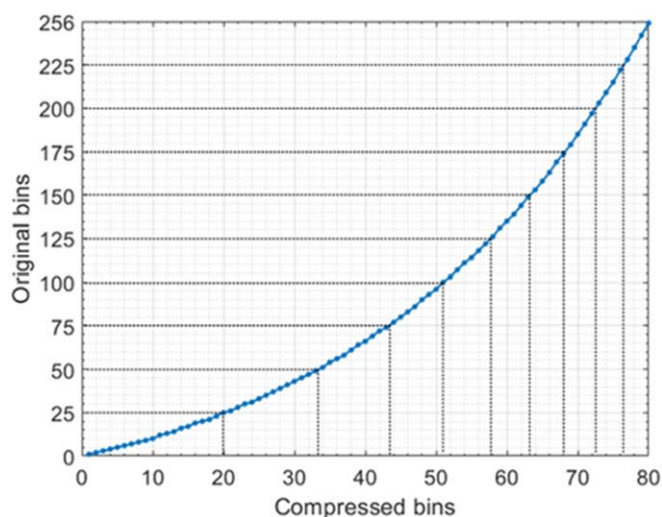
Figure 5. Nonlinear time-bin mapping, merging original 256 time-bin to compressed 80 time-bin.
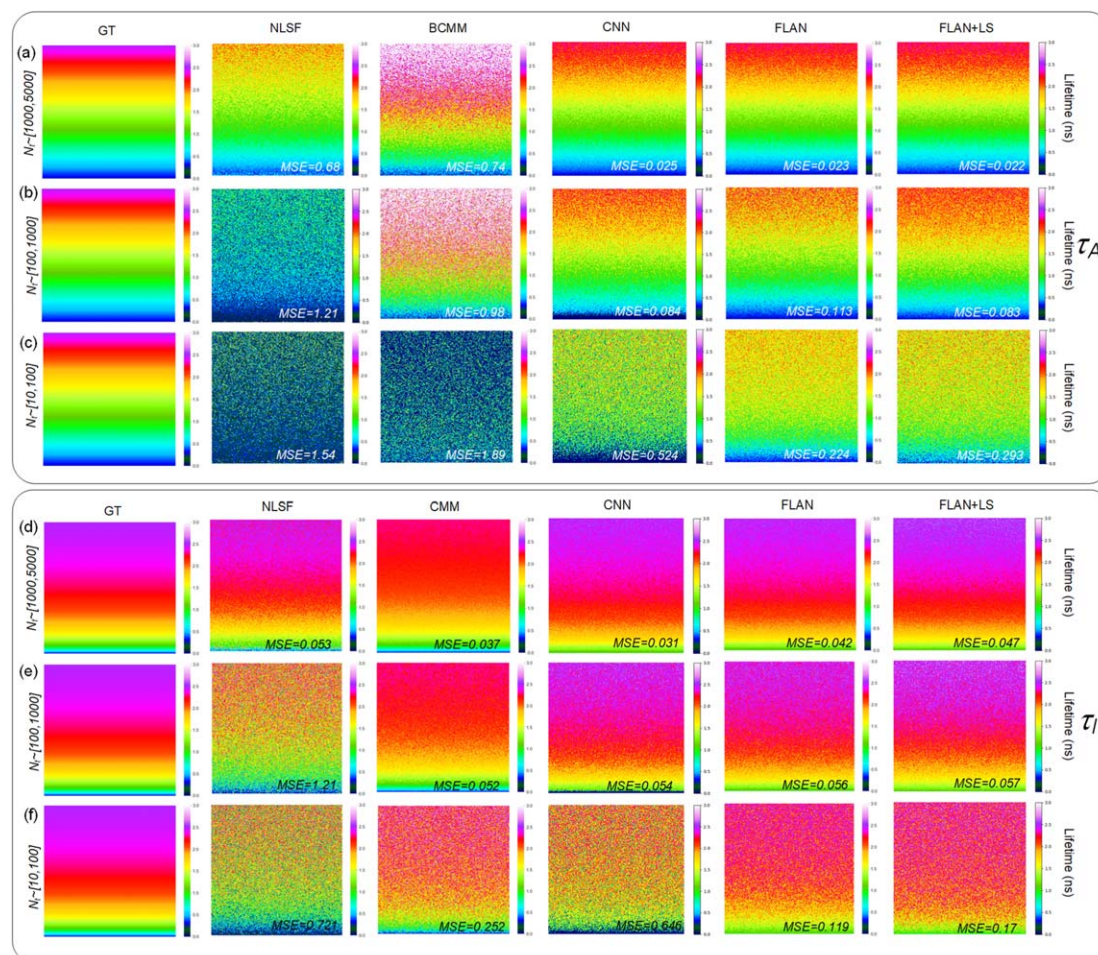

Figure 6. Synthetic GT and reconstructed τ_A and τ_I FLIM images. MSE is used to evaluate the accuracy. (a), (b), and (c) reconstructed τ_A in [1000, 5000], [100, 1000], and [10, 100] photon-count. (d), (e), and (f) reconstructed τ_I in [1000, 5000], [100, 1000], and [10, 100] photon-count.

10 × objective lens (N.A = 0.25) with a 685 nm short pass filter. Our setup acquired both sets of beads using LSM 510 confocal laser scanning microscope. A TCSPC (SPC-830, Becker & Hickl GmbH) card was mounted on the computer with a PCIe interface to record the time-of-flight of fluorescence emission. The spatial resolution of the FLIM data was configured as 256 × 256 on the SPImage software, and the field of
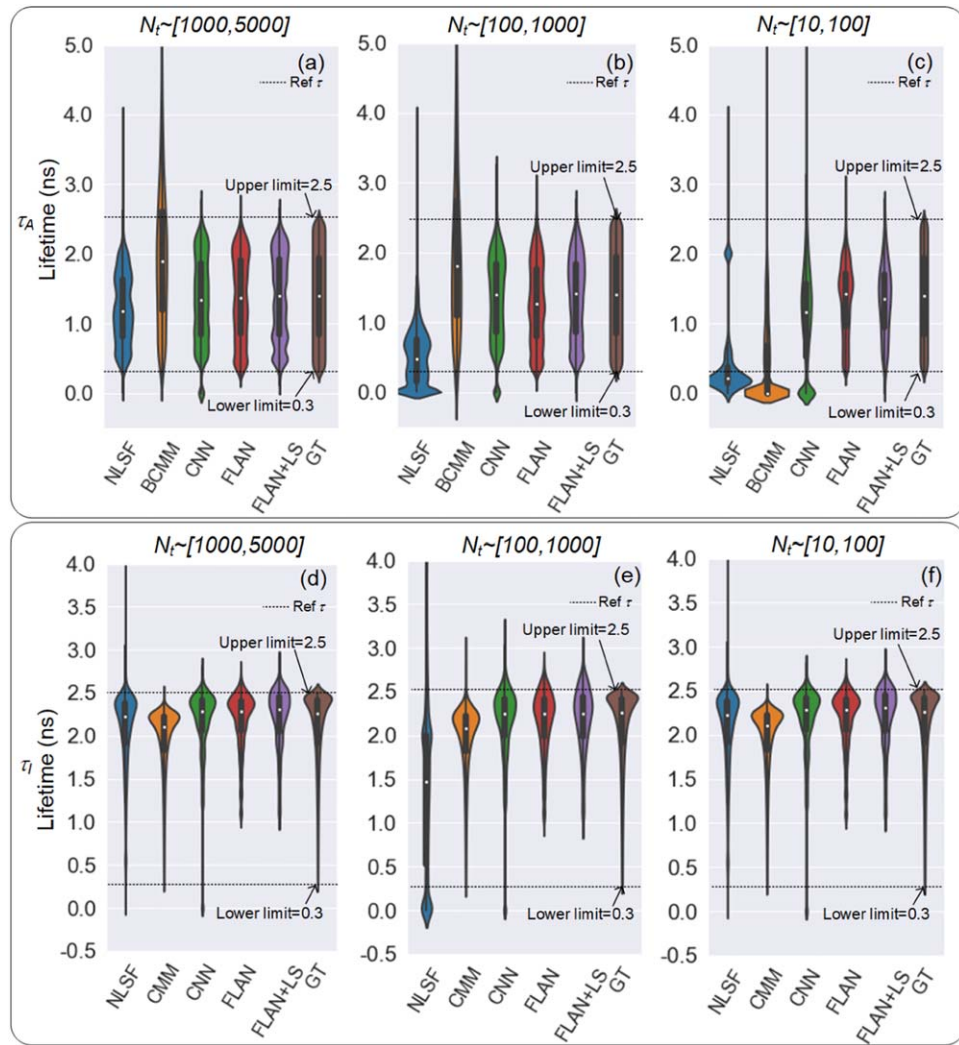
Figure 7. Lfetimes' distributions of τ_A in (a), (b), and (c); τ_I in (d), (e), and (f) retrieved from different algorithms in different levels of photon count.

view was selected by moving the translation stage. The number of time-bin was configured as 256 in each pixel, and 39.06 ps was the temporal resolution in each time-bin.

7.3. Quantitative analysis

Given that phasor [45] is a fitting-free, frequency-domain approach to observing lifetime distributions represented by polar coordinates *g* and *s*
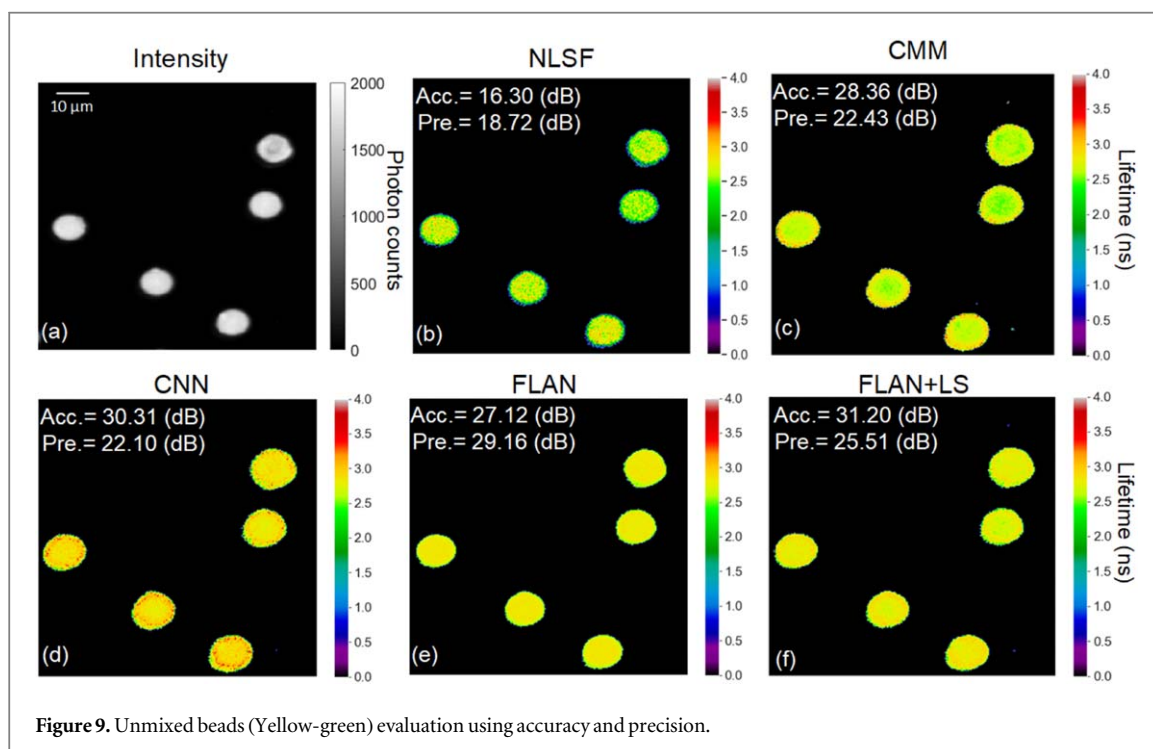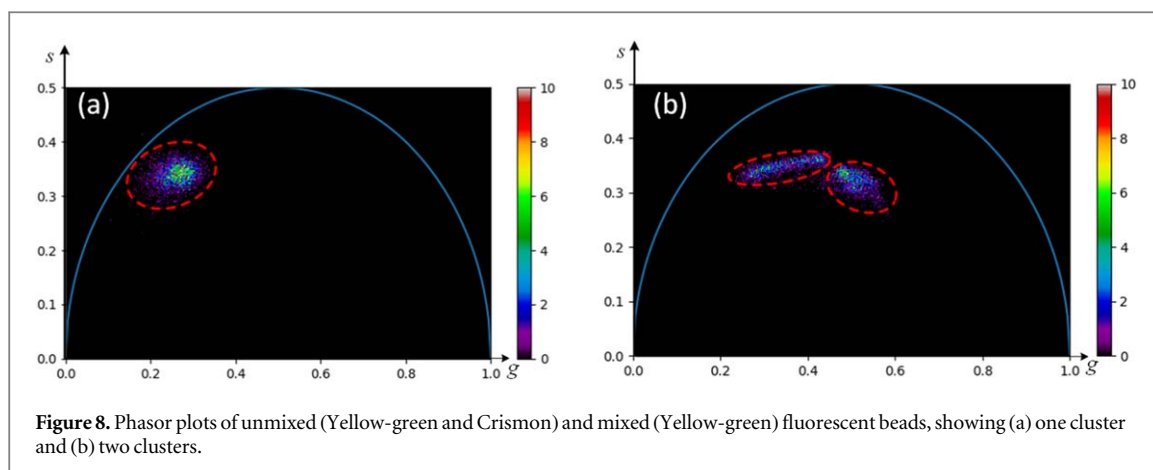
$$\begin{cases} g_{i,j} = \dfrac{\int_0^\infty h(t)_{i,j}\cos(kwt)\,dt}{\int_0^\infty h(t)_{i,j}\,dt} \\[4mm] s_{i,j} = \dfrac{\int_0^\infty h(t)_{i,j}\sin(kwt)\,dt}{\int_0^\infty h(t)_{i,j}\,dt} \end{cases}, \qquad (9)$$

where *i* and *j* is the pixels' index in the FLIM image, *kw* is the angular frequency of the excitation source. Here, we initially used a phasor plot to analyze distributions of lifetimes in each pixel to validate the prepared beads. As shown in figure 8, the cluster is not

located in the semi-circle, meaning that the two types of beads are not perfect mono- and bi-exponential decays resulting from a mixture of dyes. Given that there is no spectral overlap of the beads, meaning no energy transfer occurs, we used τ_I to analyze reconstructed lifetime images and to compare different algorithms. Two metrics, i.e. accuracy and precision in *dB* [25], were used for evaluation, defined as

$$\begin{cases} Acc. = 20\log_{10} mean(\hat{\tau}_I/\Delta\tau_I) \\ Pre. = 20\log_{10} mean(\hat{\tau}_I/\sigma\tau_I) \end{cases}, \qquad (10)$$

where $\hat{\tau}_I$ is reconstructed lifetimes, $\Delta\tau_I$ and $\sigma\tau_I$ means absolute error and standard deviation of the reconstructed lifetimes. From figure 9, we can tell DL methods obtained similar accuracy versus CMM, yet FLAN and FLAN + LS exhibit the best precision (small standard deviations). Note that NLSF has an obvious bias compared to others because it involves iterative deconvolutions that significantly rely on initial values. Likewise, we used τ_I to evaluate the algorithms for mixed beads. According to figure 8(b), as there were two individual clusters, we applied different

Figure 8. Phasor plots of unmixed (Yellow-green and Crismon) and mixed (Yellow-green) fluorescent beads, showing (a) one cluster and (b) two clusters.

Figure 9. Unmixed beads (Yellow-green) evaluation using accuracy and precision.
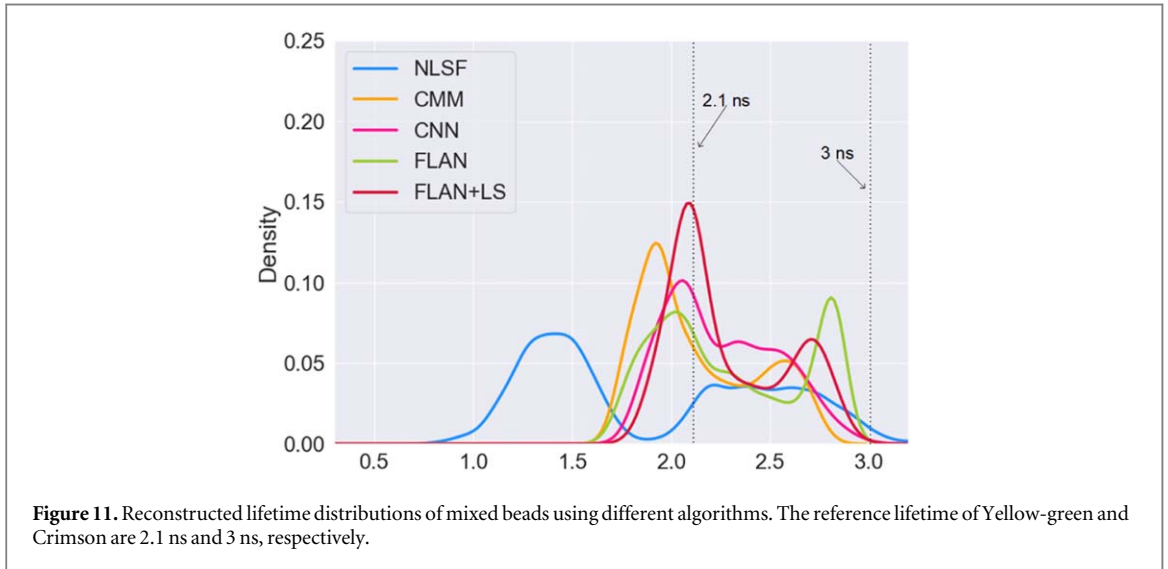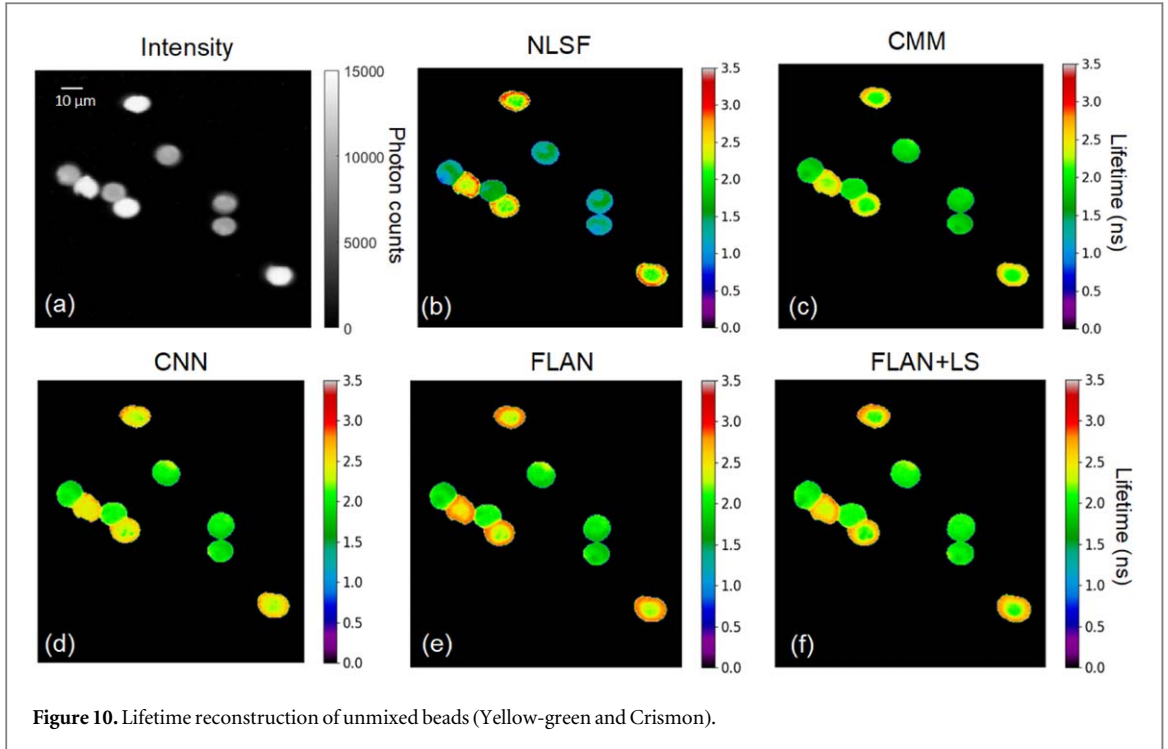
algorithms to discriminate between these two types of beads. Reconstructed FLIM images and lifetime distributions from each algorithm were plotted in figures 10 and 11. The CNN is less discerning than FLAN and FLAN + LS to discriminate between the two types of beads. CMM can distinguish the beads well, but bias still exists. Peaks' indexes of FLAN and FLAN + LS are closest to the reference lifetime. Hence, our networks resolve two types of beads with accuracy lifetime estimation. Figure 12 shows reconstructed lifetime results for low counts condition where the maximum count is 200. Figure 12(b) is the phasor projection image calculated from high counts decays (with maximum counts of 2500) with a longer acquisition time (15 seconds). Phasor projection images were leveraged as a reference to investigate the contrast of lifetime images [28, 36]. Figures 12(c) and (d) indicate reconstructed τ_A and τ_I from FLAN; figures 12(e) and (f) indicate reconstructed τ_A and τ_I from FLAN + LS. FLAN and FLAN + LS exhibit similar contrast

compared to high-count phasor protection image, meaning that our algorithms are also robust in real low-count conditions.

8. Hardware implementation

The FLAN and FLAN + LS were implemented on a PYNQ-Z2 board integrating the Processing System (PS) and the Programmable logic (PL). The DL cores were implemented in the PL part using HLS. Hardware drivers of the cores and less computing-intensive arithmetic, such as eliminating background pixels and decoding input FLP histograms into FP, were programmed in the PS. As shown in figure 13, four DL cores were instantiated to process four pixels concurrently to achieve instances-level parallel computing. The four consecutive histograms from an AXI- bus (through a high-performance slave port) should be

Figure 10. Lifetime reconstruction of unmixed beads (Yellow-green and Crismon).

Figure 11. Reconstructed lifetime distributions of mixed beads using different algorithms. The reference lifetime of Yellow-green and Crimson are 2.1 ns and 3 ns, respectively.

copied into four memory buffers. And then, the four chunks of data will be fed into corresponding DL cores. Learned parameters, including weights and BN, were fetched from the pre-trained model and exported into BRAMs using an automatic Python script. Thanks to the quantization strategy, all the parameters encoded in shortened bit-width save memories and can be pre-stored in on-chip BRAMs instead of off-chip DDR memory, which decreases the latency of data movement. Before computing lifetimes, according to the pre-defined threshold, a binarized mask map with zeros and ones should be generated offline. Only the pixels' indexes with ones will be processed. Similar to learned parameters, original input histograms should also be encoded from FLP to FP. This encoding process was conducted in the ARM processor. Eventually, the output lifetimes will be converted to readable FLP after

life-times inference. As mentioned in section IV A, gathered AC, ReLU, and BN can simplify the hardware implementation; we merge the three modules into one hardware module, indicated by grey areas next to the main AC modules in figure 14(a). The original channel-wise BN operation in i^{th} output channel of forwards-propagation is

$$x_{BN} = \frac{(x(i) - \mu(i))}{\sqrt[2]{\sigma(i)^2 + \varepsilon}} \gamma(i) + \beta(i), \tag{11}$$

where γ and β are learned parameters that can be parsed from pre-trained models, μ and σ^2 are the statistical mean and standard deviation of x, ε is a constant to avoid dividing by zero (0.0001 by default). The scaling and shift coefficients can be extracted as

Figure 12. Reconstructed lifetime images of Convallaria majalis cells. (a) intensity image; (b) phasor projection image; (c) and (d) reconstructed τ_A and τ_I from FLAN; (e) and (f) reconstructed τ_A and τ_I from FLAN + LS.

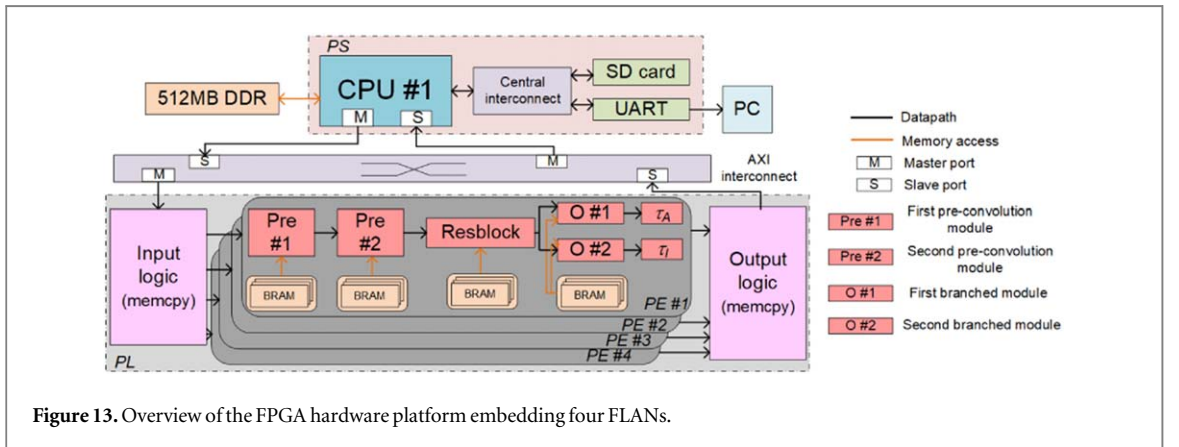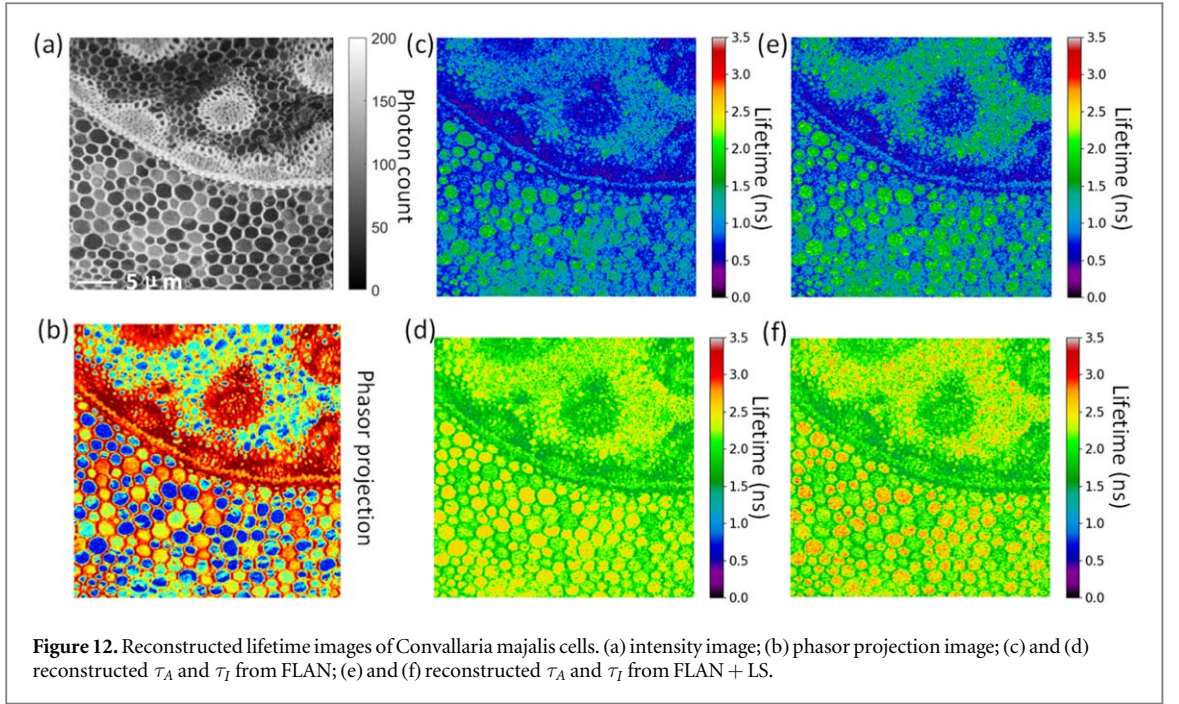**Figure 13.** Overview of the FPGA hardware platform embedding four FLANs.

Table 3. Comparisons of hardware-efficient DL networks implemented on FPGA.
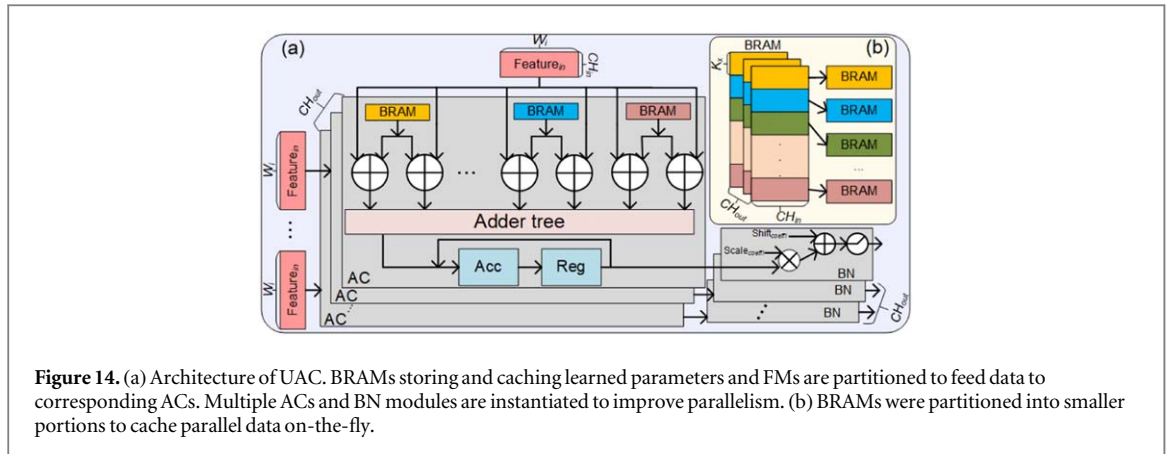
	1D CNN (2-core)	FLAN (4-core)	FLAN + LS (4-core)
Frequency	98 MHz	110 MHz	120 MHz
Precision	32-b FLP	16-b FP	16-b FP
DSP (220 Ava.)	51 (23.2%)	100 (45.4%)	76 (34.6%)
LUT (53.2K Ava.)	51 072 (95.8%)	25 886 (48.6%)	18 553 (34.9%)
FF (106.3K Ava.)	30 827 (28.7%)	18 816 (17.7%)	15 629 (14.7%)
BRAM (140 Ava.)	41 (29.1%)	124.5 (88.9%)	47 (33.6%)
LUTRAM (17.4 K Ava.)	1566 (9%)	2202 (12.7%)	2198 (12.6%)
τ_A Accuracy (MSE)	0.121 ns	0.125 ns	0.138 ns
τ_I Accuracy (MSE)	0.089 ns	0.083 ns	0.102 ns
Power (W)	2.56	1.93	1.78
Latency (ms)	1.73	0.16	0.11
Throughout (PMS)	15	33	40
Efficiency (PMS/W)	5.86	17.10	22.47

$$\begin{cases} scale(i) = \dfrac{\gamma(i)}{\sqrt[2]{\sigma(i)^2 + \varepsilon}} \\ shift(i) = \beta(i) - \dfrac{\gamma(i)\mu(i)}{\sqrt[2]{\sigma(i)^2 + \varepsilon}} \end{cases}. \qquad (12)$$

Therefore, BN can be simplified as matrix multiplications and vector additions on the hardware

$$x_{BN} = scale(i) \times x(i) + shift(i), \qquad (13)$$

scale and *shift* were calculated and extracted offline using a Python script. As limited writing and reading

Figure 14. (a) Architecture of UAC. BRAMs storing and caching learned parameters and FMs are partitioned to feed data to corresponding ACs. Multiple ACs and BN modules are instantiated to improve parallelism. (b) BRAMs were partitioned into smaller portions to cache parallel data on-the-fly.

Figure 15. Relationships between batch sizes and throughput (pixel/ms) in the inference phase on CPU, GPU, and FPGA + CPU.

ports of BRAM would hinder simultaneous data transfer to multiple PEs. To overcome the challenge, BRAMs under usage were partitioned into small portions to increase write-read bandwidth, leading to more hardware consumption. Due to our models' compaction, all the intermediate results among layers were cached in on-chip BRAM instead of accessing the DDR memory. After processing four adjacent pixels, only a vector containing eight inferred lifetime values will be output to ARM via address-mapping AXI for anti-quantization decoding and validation. Table 3 compares hardware utilization and computing performance of three pixel-wise DL networks, namely, 1D CNN with 32-b FLP, FLAN with 16-b FP, and FLAN + LS with 16-b FP, implemented on FPGA. The accuracy was assessed by MSE using synthetic test datasets. We randomly picked 100 histograms and divided them into 25 batches as our DL processors can process 4 pixels concurrently. And the batches of histograms were stored in the DDR. For 1D CNN, only 2 DL cores can be accommodated due to the FLP data format, whereby the 100 histograms should be divided into 50 batches. The latency of the three networks is presented by the consumed time for processing 4 pixels. Throughput was defined by

processed pixels per millisecond (PMS). MSE was adopted to assess the accuracy, aligning with figure 6. The hardware utilization was obtained after place-and-route implementation. The power consumption was estimated in the post-implementation phase using power analyzer tools in Vivado 2018.2. FLAN and FLAN + LS with shorter bit-width save multitudes of hardware and achieve high operating frequency. FLAN and FLAN + LS still utilize DSPs as multiplications of BN were synthesized using DSP. Although four DL units were implemented in FLAN + LS, it consumes the fewest look-up-table (LUT) and flip-flops (FF). Due to the numerous partitioned memory blocks in FLAN and FLAN + LS, more distributed (LUT) RAMs were consumed to meet the memory bandwidth. Despite the on-chip quantization, there was no apparent accuracy degradation for τ_A and τ_I reconstruction.

We also investigated how batch sizes would affect the performance of FLAN + LS on CPU, GPU, and FPGA, depicted in figure 15. GPU's computing performance significantly relied on the batch size and the scale of DL models. Small batch size and model size would dramatically deteriorate the speed. The figure

shows that CPU and GPU present similar throughput when the batch size is smaller than 32. GPU's throughput exceeds CPU when the batch size exceeds 32 and FPGA when the batch size is more significant than 128. Note that, for some FLIM microscope applications like ours, the contrast between the sample and background should be distinguished, whereby each pixel should be compared to the threshold. This compels GPU process histograms pixel-by-pixel in serial (batch size equals 1) instead of parallel. Moreover, pixel-wise data movement on a GPU consumes extra time. Another reason hindering GPU's performance is that CUDA and cuDNN are not providing optimizations for adder-based convolutions. As for FPGAs, they are not restricted by batch sizes, and they excel at processing data streams and addition operations.

9. Conclusion

This study reports a multiplication-free and robust DL network for time-domain FLIM analysis. Besides, two data compression strategies were introduced to accelerate the processing. Both synthetic and real data were adopted to extensively and quantitatively demonstrate superior results comparing existing algorithms. Besides, to illustrate the applicability of hardware, we implemented our DL architectures on FPGA and compared them with traditional 1D CNN on hardware. Results show that our compression methods can significantly save hardware resources and achieve high computational efficiency without performance deterioration. Also, to investigate our lightweight DL network's performance on GPU, CPU, and FPGA + CPU. We conclude that our heterogenous FPGA + CPU is not affected by batch sizes and outperforms GPU for small batch sizes. A potential improvement of this work is that the training datasets and the network architecture are tailored for confocal microscopes that require a scanning mechanism and long acquisition time to guarantee high SNR of the captured data. Whereas for wide-field FLIM, imaging systems using SPAD array do not use scanning strategies but wide-field illumination, leading to shorter acquisition time and low SNR. And IRFs vary for different systems. Therefore, training datasets and the IRF should be modified to cater to various single-photon sensors. Further, our DL hardware cores cannot be integrated with the TCSPC module coupled with the confocal microscope due to the non-configurable property of the TCSPC board. However, it is possible to embed the DL cores in the data readout circuit of SPAD arrays because FPGA acts as the readout and control hardware for SPAD arrays. We will make our DL more generalized for SPAD array-based systems in software and hardware aspects.

Apart from FLIM, the DL network can be applied to analyze data from diffuse correlation spectroscopy (DCS) [46] that also relies on single-photon sensors and TCSPC systems. For DCS, the essential task is to retrieve scattering coefficients and blood flow index from temporal correlation functions. To migrate the workflow in this work to DCS, we need to alter the theoretical modelling of training datasets and the optical path, and this work paves the way for DCS research.

Acknowledgments

Data availability statement

The data that support the findings of this study are available upon reasonable request from the authors. The data that support the findings of this study are available upon reasonable request from the authors.

Funding

Conflicts of interest

There are no conflicts to declare.

ORCID iDs

Zhenya Zang ⬤ https://orcid.org/0000-0003-4952-6727
Yu Chen ⬤ https://orcid.org/0000-0003-2427-3559
David Day Uei Li ⬤ https://orcid.org/0000-0002-6401-4263

References

[1] Yaseen M A, Sakadžić S, Wu W, Becker W, Kasischke K A and Boas D A 2013 *In vivo* imaging of cerebral energy metabolism with two-photon fluorescence lifetime microscopy of NADH *Biomed. Opt. Express* **4** 307–21

[2] Dancik Y, Favre A, Loy C J, Zvyagin A V and Roberts M S 2013 Use of multiphoton tomography and fluorescence lifetime imaging to investigate skin pigmentation *in vivo J. Biomed. Opt.* **18** 026022

[3] Nobis M *et al* 2013 Intravital FLIM-FRET imaging reveals dasatinib-induced spatial control of src in pancreatic cancerimaging drug targeting in live tumors *Cancer Res.* **73** 4674–86

[4] Karpf S, Riche C T, Di Carlo D, Goel A, Zeiger W A, Suresh A, Portera-Cailliau C and Jalali B 2020 Spectro-temporal encoded multiphoton microscopy and fluorescence lifetime imaging at kilohertz frame-rates *Nat. Commun.* **11** 1–9

[5] Sun Y H, Hatami N, Yee M, Phipps J E, Elson D S, Gorin F, Schrot R J and Marcu L 2010 Fluorescence lifetime imaging microscopy for brain tumor image-guided surgery *J. Biomed. Opt.* **15** 056022

[6] Alfonso-Garcia A, Bec J, Sridharan Weaver S, Hartl B, Unger J, Bobinski M, Lechpammer M, Girgis F, Boggan J and Marcu L 2020 Real-time augmented reality for delineation of surgical margins during neurosurgery using autofluorescence lifetime contrast *J. Biophotonics* **13** e201900108

[7] Becker W 2005 *Advanced Time-Correlated Single Photon Counting Techniques* (Springer Science & Business Media) vol 81

[8] Suhling K *et al* 2019 Wide-field time-correlated single photon counting-based fluorescence lifetime imaging microscopy *Nucl. Instrum. Methods Phys. Res., Sect. A* **942** 162365

[9] Thiele J C, Helmerich D A, Oleksiievets N, Tsukanov R, Butkevich E, Sauer M, Nevskyi O and Enderlein J 2020 Confocal fluorescence-lifetime single-molecule localization microscopy *ACS nano* **14** 14190–200

[10] Philip J and Carlsson K 2003 Theoretical investigation of the signal-to-noise ratio in fluorescence lifetime imaging *J. Opt. Soc. Am. A* **20** 368–79

[11] Lakowicz J R 2006 *Principles Of Fluorescence Spectroscopy* (Berlin: Springer)

[12] Bajzer Ž and Prendergast F G 1992 [10] Maximum likelihood analysis of fluorescence data *Methods Enzymol.* (Amsterdam: Elsevier) vol 210, pp 200–37

[13] Li D U, Tyndall D, Walker R, Richardson J A, Henderson R K, Arlt J, Stoppa D and Charbon E 2011 Video-rate fluorescence lifetime imaging camera with CMOS single-photon avalanche diode arrays and high-speed imaging algorithm *J. Biomed. Opt.* **16** 096012

[14] Li D D-U, Yu H and Chen Y 2015 Fast bi-exponential fluorescence lifetime imaging analysis methods *Opt. Lett.* **40** 336–9

[15] Sillen A and Engelborghs Y 1998 The correct use of 'average' fluorescence parameters *Photochem. Photobiol.* **67** 475–86

[16] Li Y, Natakorn S, Chen Y, Safar M, Cunningham M, Tian J and Li D D-U 2020 Investigations on average fluorescence lifetimes for visualizing multi-exponential decays *Front. Phys.* **8** 576862

[17] Kim B, Park B, Lee S and Won Y 2016 GPU accelerated real-time confocal fluorescence lifetime imaging microscopy (FLIM) based on the analog mean-delay (AMD) method *Biomed. Opt. Express* **7** 5055–65

[18] Sorrells J E, Iyer R R, Yang L, Bower A J, Spillman D R, Chaney E J, Tu H and Boppart S A 2021 Real-time pixelwise phasor analysis for video-rate two-photon fluorescence lifetime imaging microscopy *Biomed. Opt. Express* **12** 4003–19

[19] Ma Y, Lee Y, Best-Popescu C and Gao L 2021 High-speed compressed-sensing fluorescence lifetime imaging microscopy of live cells *Proc. Natl. Acad. Sci.* **118** e2004176118

[20] Gyongy I, Calder N, Davies A, Dutton N A, Duncan R R, Rickman C, Dalgarno P and Henderson R K 2017 A 256 × 256, 100-kfps, 61% fill-factor SPAD image sensor for time-resolved microscopy applications *IEEE Trans. Electron Devices* **65** 547–54

[21] Li D-U, Rae B R, Andrews R, Arlt J and Henderson R K 2010 Hardware implementation algorithm and error analysis of high-speed fluorescence lifetime sensing systems using center-of-mass method *J. Biomed. Opt.* **15** 017006

[22] Tyndall D, Rae B R, Li D D-U, Arlt J, Johnston A, Richardson J A and Henderson R K 2012 A high-throughput time-resolved mini-silicon photomultiplier with embedded fluorescence lifetime estimation in 0.13 μm CMOS *IEEE Trans. Biomed. Circuits Syst.* **6** 562–70

[23] Krstajić N, Levitt J, Poland S, Ameer-Beg S and Henderson R 2015 256 × 2 SPAD line sensor for time resolved fluorescence spectroscopy *Opt. Express* **23** 5653–69

[24] Della Rocca F M, Nedbal J, Tyndall D, Krstajić N, Li D D-U, Ameer-Beg S M and Henderson R K 2016 Real-time fluorescence lifetime actuation for cell sorting using a CMOS SPAD silicon photomultiplier *Opt. Lett.* **41** 673–6

[25] Xiao D, Zang Z, Sapermsap N, Wang Q, Xie W, Chen Y and Li D D U 2021 Dynamic fluorescence lifetime sensing with CMOS single-photon avalanche diode arrays and deep learning processors *Biomed. Opt. Express* **12** 3450–62

[26] Serafino M J, Applegate B E and Jo J A 2020 Direct frequency domain fluorescence lifetime imaging using field programmable gate arrays for real time processing *Rev. Sci. Instrum.* **91** 033708

[27] Wu G, Nowotny T, Zhang Y, Yu H-Q and Li D D-U 2016 Artificial neural network approaches for fluorescence lifetime imaging techniques *Opt. Lett.* **41** 2561–4

[28] Xiao D, Chen Y and Li D D-U 2021 One-dimensional deep learning architecture for fast fluorescence lifetime imaging *IEEE J. Sel. Top. Quantum Electron.* **27** 1–10

[29] Zickus V *et al* 2020 Fluorescence lifetime imaging with a megapixel SPAD camera and neural network lifetime estimation *Sci. Rep.* **10** 1–10

[30] Héliot L and Leray A 2021 Simple phasor-based deep neural network for fluorescence lifetime imaging microscopy *Sci. Rep.* **11** 1–13

[31] Smith J T, Yao R, Sinsuebphon N, Rudkouskaya A, Un N, Mazurkiewicz J, Barroso M, Yan P and Intes X 2019 Fast fit-free analysis of fluorescence lifetime imaging via deep learning *Proc. Natl. Acad. Sci.* **116** 24019–30

[32] Ochoa M, Rudkouskaya A, Yao R, Yan P, Barroso M and Intes X 2020 High compression deep learning based single-pixel hyperspectral macroscopic fluorescence lifetime imaging *in vivo Biomed. Opt. Express* **11** 5401–24

[33] Chen Y-I *et al* 2022 Generative adversarial network enables rapid and robust fluorescence lifetime image analysis in live cells *Communications Biology* **5** 1–11

[34] Smith J T, Ochoa M and Intes X 2020 UNMIX-ME: spectral and lifetime fluorescence unmixing via deep learning *Biomed. Opt. Express* **11** 3857–74

[35] Xiao D, Zang Z, Xie W, Sapermsap N, Chen Y and Li D D U 2022 Spatial resolution improved fluorescence lifetime imaging via deep learning *Opt. Express* **30** 11479–94

[36] Zang Z, Xiao D, Wang Q, Li Z, Xie W, Chen Y and Li D D U 2022 Fast analysis of time-domain fluorescence lifetime imaging via extreme learning machine *Sensors* **22** 3758

[37] Hubara I, Courbariaux M, Soudry D, El-Yaniv R and Bengio Y 2016 Binarized neural networks *Advances in Neural Information Processing Systems* **29**

[38] Wang Y, Huang M, Han K, Chen H, Zhang W, Xu C and Tao D 2021 AdderNet and its minimalist hardware design for energy-efficient artificial intelligence arXiv:2101.10015

[39] Chen H, Wang Y, Xu C, Shi B, Xu C, Tian Q and Xu C 2020 AdderNet: Do we really need multiplications in deep learning? *Proc. of the IEEE/CVF conference on computer vision and pattern recognition* pp 1468–77

[40] He K, Zhang X, Ren S and Sun J 2016 Deep residual learning for image recognition *Proc. of the IEEE conference on computer vision and pattern recognition* pp 770–8

[41] Guo K, Zeng S, Yu J, Wang Y and Yang H A 2018 Survey of FPGA-based neural network inference accelerator Arxiv: arXiv:1712.08934

[42] Trinh A L and Esposito A 2021 Biochemical resolving power of fluorescence lifetime imaging: untangling the roles of the instrument response function and photon-statiikstics *Biomed. Opt. Express* **12** 3775–88

[43] Kage D, Hoffmann K, Nifontova G, Krivenkov V, Sukhanova A, Nabiev I and Resch-Genger U 2020 Tempo-spectral multiplexing in flow cytometry with lifetime detection using QD-encoded polymer beads *Sci. Rep.* **10** 1–11

[44] Bares A J *et al* 2020 Hyperspectral multiphoton microscopy for *in vivo* visualization of multiple, spectrally overlapped fluorescent labels *Optica* **7** 1587–601

[45] Digman M A, Caiolfa V R, Zamai M and Gratton E 2008 The phasor approach to fluorescence lifetime imaging analysis *Biophys. J.* **94** L14–6

[46] Tamborini D, Stephens K A, Wu M M, Farzam P, Siegel A M, Shatrovoy O, Blackwell M, Boas D A, Carp S A and Franceschini M A 2019 Portable system for time-domain diffuse correlation spectroscopy *IEEE Trans. Biomed. Eng.* **66** 3014–25