FAST FLUORESCENCE LIFETIME IMAGING AND SENSING VIA DEEP LEARNING

DONG XIAO

A thesis submitted for the degree of Doctor of Philosophy

THE UNIVERSITY OF STRATHCLYDE

October 2022

This thesis is the result of the author’s original research. It has been composed by the author and has not been previously submitted for examination which has led to the award of a degree.

Signed: Dong Xiao Date:21 October 2022

Abstract

Fluorescence lifetime imaging microscopy (FLIM) has become a valuable tool in diverse disciplines. This thesis presents deep learning (DL) approaches to addressing two major challenges in FLIM: slow and complex data analysis and the high photon budget for precisely quantifying the fluorescence lifetimes. DL's ability to extract high-dimensional features from data has revolutionized optical and biomedical imaging analysis. This thesis contributes several novel DL FLIM algorithms that significantly expand FLIM's scope.

Firstly, a hardware-friendly pixel-wise DL algorithm is proposed for fast FLIM data analysis. The algorithm has a simple architecture yet can effectively resolve multi-exponential decay models. The calculation speed and accuracy outperform conventional methods significantly.

Secondly, a DL algorithm is proposed to improve FLIM image spatial resolution, obtaining high-resolution (HR) fluorescence lifetime images from low-resolution (LR) images. A computational framework is developed to generate large-scale semi-synthetic FLIM datasets to address the challenge of the lack of sufficient high-quality FLIM datasets. This algorithm offers a practical approach to obtaining HR FLIM images quickly for FLIM systems.

Thirdly, a DL algorithm is developed to analyze FLIM images with only a few photons per pixel, named Few-Photon Fluorescence Lifetime Imaging (FPFLI) algorithm. FPFLI uses spatial correlation and intensity information to robustly estimate the fluorescence lifetime images, pushing this photon budget to a record-low level of only a few photons per pixel.

Finally, a time-resolved flow cytometry (TRFC) system is developed by integrating an advanced CMOS single-photon avalanche diode (SPAD) array and a DL processor. The SPAD array, using a parallel light detection scheme, shows an excellent photon-counting throughput. A quantized convolutional neural network (QCNN) algorithm is designed and implemented on a field-programmable gate array as an embedded processor. The processor resolves fluorescence lifetimes against disturbing noise, showing unparalleled high accuracy, fast analysis speed, and low power consumption.

Acknowledgement

I would like to express my sincere gratitude to everyone who contributed to the success of my research and the completion of this thesis.

First and foremost, I am grateful to my supervisors at the University of Strathclyde, Dr David Li and Dr Yu Chen. Their continuous guidance, encouragement, and support were instrumental in the completion of this research. I particularly want to express my special thanks to David for being an exceptional supervisor throughout my PhD journey. Without his profound knowledge and tireless dedication, this achievement would not have been possible.

I also want to acknowledge the generous support of Medical Research Scotland (MRS) for providing the necessary funding for my research. I am also thankful to Dr Richard Walker, the CEO of Photon Force Ltd., for his financial and technical assistance.

I am indebted to my colleagues who directly and indirectly supported my research. I am particularly grateful to Haochang Chen, who guided me in building the SPAD systems and configuring the firmware and software. Zhenya Zang provided me with technical support for hardware design and implementation. Natakorn Sapermsap and Quan Wang helped me develop the flow cytometry system and FLIM measurements. I would also like to thank Prof. Robert K. Henderson's research group at the University of Edinburgh for providing me with cutting-edge SPAD chips and firmware. Special thanks also to my research group colleagues for their invaluable support and friendship throughout my research.

Finally, I want to thank my family for their unwavering support throughout my PhD journey. To my wife Qian, thank you for your love, patience, and support, which made it possible for me to focus on my research. I am especially grateful for your sacrifice of four years to be with me throughout this journey. To my little Dennis, you are the light of my life, and your adorable distractions during my research were a sweet burden that kept me motivated. I dedicate this thesis to both of you, and I am forever grateful for your love and support.

Contents

List of Figures

List of Tables

List of Abbreviations

AI	**A**rtificial **I**ntelligence
ANN	**A**rtificial **N**eural **N**etwork
ASIC	**A**pplication-**S**pecific **I**ntegrated **C**ircuit
BA	**B**ayesian **A**pproach
BN	**B**atch **N**ormalization
BP	**B**ack**p**ropagation
BRAM	**B**lock **R**andom-**A**ccess **M**emory
CAP	**C**redit **A**ssignment **P**roblem
CLS	**C**onfocal **L**aser **S**canning
CMM	**C**enter-of-**M**ass **M**ethod
CNN	**C**onvolutional **N**eural **N**etwork
CMOS	**C**omplementary **M**etal-**o**n-**S**emiconductor
DAG	**D**irected **A**cyclic **G**raph
DL	**D**eep **L**earning
DMA	**D**irect-**M**emory-**A**ccess
DMEM	**D**ulbecco’s **M**odified **E**agle **M**edium
DNN	**D**eep **N**eural **N**etwork
DOF	**D**epth-**o**f-**F**ield
EGFP	**E**nhanced **G**reen **F**luorescent **P**rotein
ELU	**E**xponential **L**inear **U**nit
FCS	**F**etal **C**alf **S**erum
FD	**F**requency **D**omain
FFPE	**F**ormalin-**F**ixed and **P**araffin-**E**mbedded
FLIM	**F**luorescence **L**ifetime **I**maging **M**icroscopy
FLOP	**Fl**oating-**P**oint **O**peration
FNN	**F**eedforward **N**eural **N**etwork
FOV	**F**ield-**o**f-**V**iew
FPGA	**F**ield-**P**rogrammable **G**ate **A**rray
FRET	**F**örster **R**esonance **E**nergy **T**ransfer

FSC	**F**orward **Sc**atter
FWHM	**F**ull **W**idth at **H**alf **M**aximum
GAN	**G**enerative **A**dversarial **N**etwork
GELU	**G**aussian **E**rror **L**inear **U**nit
GPU	**G**raphics **P**rocessing **U**nit
GRU	**G**ated **R**ecurrent **U**nit
GT	**G**round **T**ruth
H&E	**H**aematoxylin and **E**osin
HLS	**H**igh-**L**evel-**S**ynthesis
HPA	**H**uman **P**rotein **A**tlas
HR	**H**igh-**R**esolution
ICM	**I**terated **C**onditional **M**odel
IEM	**I**ntegral **E**xtraction **M**ethod
INR	**I**mplicit **N**eural **R**epresentation
IRF	**I**nstrument **R**esponse **F**unction
LiDAR	**L**ight **D**etection **a**nd **R**anging
LIIF	**L**ocal **I**mplicit **I**mage **F**unction
LLE	**L**ocal **L**ifetime **E**stimator
LR	**L**ow-**R**esolution
LSF	**L**east **S**quare **F**itting
LSF-LE	**L**east **S**quares **F**itting **B**ased on **L**aguerre **E**xpansion
LSM	**L**aser **S**canning **M**icroscopy
LSTM	**L**ong **S**hort-**T**erm **M**emory Network
MAE	**M**ean **A**bsolute **E**rror
MCMC	**M**arkov **C**hain **M**onte **C**arlo
ML	**M**achine **L**earning
MLE	**M**aximum **L**ikelihood **E**stimation
MLP	**M**ulti-**L**ayer **P**erceptron
MLS	**M**ultiphoton **L**aser **S**canning
MRF	**M**arkov **R**andom **F**ield
MSE	**M**ean **S**quare **E**rror
MS-SSIM	**M**ulti**s**cale **S**tructural **S**imilarity **I**ndex **M**easure
MTF	**M**odulation **T**ransfer **F**unction

NII	**N**eural **I**mplicit **I**nterpolation
NN	**N**eural **N**etwork
OTF	**O**ptical **T**ransfer **F**unction
PA	**P**hasor **A**nalysis
PALM	**P**hoto-**A**ctivated **L**ocalization **M**icroscopy
PBS	**P**hosphate-**B**uffered **S**aline
PE	**P**rocessing **E**lement
PMT	**P**hotomultiplier
PPP	**P**hoton-**P**er-**P**ixel
PSF	**P**oint **S**pread **F**unction
PSNR	**P**eak **S**ignal-to-**N**oise **R**atio
PTF	**P**hase **T**ransfer **F**unction
QCNN	**Q**uantised **C**onvolutional **N**eural **N**etwork
ReLU	**Re**ctified **L**inear **U**nit
ResNet	**Res**idual **N**eural **Net**work
RLD	**R**apid **L**ifetime **D**etermination
RNN	**R**ecurrent **N**eural **N**etwork
ROI	**R**egions **o**f **I**nterest
SISR	**S**ingle **I**mage **S**uper **R**esolution
SGD	**St**ochastic **G**radient **D**escent
SNR	**S**ignal-to-**N**oise **R**atio
SoC	**S**ystem-**on**-**C**hip
SPAD	**S**ingle-**P**hoton **A**valanche **D**iode
SPD	**S**ingle-**P**hoton **D**etector
SSC	**S**ide **Sc**atter
STORM	**S**tochastic **O**ptical **R**econstruction **M**icroscopy
TCSPC	**T**ime-**C**orrelated **S**ingle **P**hoton **C**ounting
TD	**T**ime **D**omain
TDC	**T**ime-to-**D**igital **C**onverter
TG	**T**ime-**G**ated
TRFC	**T**ime-**R**esolved **F**low **C**ytometry
WFIM	**W**ide-**F**ield **I**llumination **M**icroscopy

List of Publications

Peer-Reviewed Journal Articles

1. **D. Xiao**, Z. Zang, W. Xie, N. Sapermsap, Y. Chen, and D. Li, "Deep learning-based single image super-resolution for fluorescence lifetime imaging microscopy", *Optics Express*, 30(7),11479-11494, (2022)

2. **D. Xiao**, Y. Chen, and D. Li, "One-dimensional deep learning architecture for fast fluorescence lifetime imaging", *IEEE Journal of Selected Topics in Quantum Electronics*, 27(4), [7000210], (2021).

3. **D. Xiao**, Z. Zang, N. Sapermsap, Q. Wang, W. Xie, Y. Chen, and D. Li, "Dynamic fluorescence lifetime sensing with CMOS single-photon avalanche diode arrays and deep learning processors", *Biomedical Optics Express*, 12(6), 3450-3462, (2021).

4. **D. Xiao**, N. Sapermsap, M. Safar, M. R. Cunningham, Y. Chen, and D. Li, "On synthetic instrument response functions of time-correlated single-photon counting based fluorescence lifetime imaging analysis", *Frontiers in Physics*, 9, [635645], (2021).

5. Z. Zang, **D. Xiao**, and D. Li, "Non-fusion time-resolved depth image reconstruction using a highly efficient neural network architecture", *Optics Express*, 29(13), 19278-19291. (2021).

6. Z., Zang, **D. Xiao**, Q. Wang, Z. Li, W. Xie, Y. Chen, and D. Li, "Fast analysis of time-domain fluorescence lifetime imaging via extreme learning machine", *Sensors*, *22*(10), 3758, (2022)

7. Q. Wang; Y. Li, **D. Xiao**, Z. Zang, Z. Jiao, Y. Chen; D. Li, "Simple and Robust Deep Learning Approach for Fast Fluorescence Lifetime Imaging", *Sensors*, 22, 7293, (2022)

8. Z. Zang, **D. Xiao**, Q. Wang, Z. Jiao, Y. Chen, and D. Li, "Compact and robust deep learning architecture for fluorescence lifetime imaging and FPGA implementation", *Methods Appl. Fluoresc.* 11, 025002 (2023).

9. Z., Jiao, Z. Zang, Q. Wang, Y. Chen, **D. Xiao**, and D. Li, "PAIM (πM): portable AI-enhanced fluorescence microscope for real-time target detection", *Optics and Laser Technology,* 163, 109356 (2023).

Preprint Manuscript

1. **D. Xiao**, Y. Chen, and D. Li "Deep learning Enhanced Fast Fluorescence Lifetime imaging with a few photons", bioRxiv, https://doi.org/10.1101/2023.04.06.534322, accepted by *Optica*.

Conference Papers

2. **D. Xiao**, Z. Zang, Q. Wang, Z. Jiao, F. M. Della Rocca, Y. Chen, and D. Li, "Smart Wide-Field Fluorescence Lifetime Imaging System with CMOS Single-Photon Avalanche Diode Arrays", 44th Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC 2022), Glasgow, UK

3. Z. Zang, **D. Xiao**, Q. Wang, Z. Jiao, Z. Li, Yu Chen, and D. Li, "Hardware Inspired Neural Network for Efficient Time-Resolved Biomedical Imaging", 44th Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC 2022), Glasgow, UK

Invited Book Chapter

1. Y. Li, L. Liu, **D. Xiao**, H. Li, N. Sapermsap, J. Tian, Y. Chen, and D. Li, "Lifetime determination algorithms for time-domain fluorescence lifetime imaging: a review", Fluorescence Imaging - Recent Advances and Applications, IntechOpen, ISBN 978-1-80355-184-5 (2022)

Chapter 1

Introduction

1.1 Research motivations and aims

In the 17th century, Antonie van Leeuwenhoek (1632–1723) used single-lens light microscopes to observe microstructures and microorganisms for the first time, opening the door to the microscopic world for humankind. Since then, the light microscope has been an indispensable tool for scientific research, contributing significantly to a wide range of modern disciplines, including biology, physics, chemistry, and pharmacy, among others. As light is non-invasive, non-destructive, and non-ionizing, a light microscope proves us an ideal route to investigate microorganisms and living cells. Most of our knowledge of biomedicine is directly acquired through observation from microscopies [1, 2].

One of the most significant challenges in light microscopy is the low contrast of biological samples. The image contrast typically comes from the scatter, absorption, and reflectance of the incident light. However, biological specimens are transparent in the visible light range and their refractive index is close to that of water. As a result, the image contrast is inherently low, leading to a poor signal-to-noise ratio (SNR) and hampering the observation of the samples. To address this issue, various methods have been developed to improve image contrast, such as using phase contrast, differential interference contrast, and staining specimens with high-contrast materials [2].

Fluorescence microscopy has revolutionized light microscopy by using fluorescence emission from the sample itself or stained dyes to provide image contrast [3]. Compared to conventional light microscopies, fluorescence microscopy has four distinct advantages:

(a) It has a better SNR. The fluorescence emission provides a much stronger signal, resulting in better image quality and a sharper distinction from the background.

(b) It is more specific. Specific cellular organelles or contents, such as nuclei, mitochondria, DNA/RNA, and molecules, can be stained with different dyes for various purposes [4].

(c) It has excellent sensitivity. Modern single-molecular imaging fluorescence microscopy uses fluorescent markers to detect and track single molecules, visualizing and analyzing single-molecule activities without changing the physiological conditions of the biological system [5].

(d) It is multifunctional. Fluorescent dyes can serve as versatile molecular probes to read out molecular dynamics and biophysical parameters of surrounding environments [6, 7].

Fluorescence has many properties, including intensity, polarization, lifetime, and absorption/emission spectra, that can be used for investigating specimens. Among these properties, fluorescence lifetime has attracted enormous attention in past decades due to its unique features. Fluorescence lifetime is the average lifetime of the excited fluorophores (fluorescent molecules) returning to the ground state. It is sensitive to fluorophores' microenvironments and their interactions with other molecules. Therefore, fluorescence lifetime is an excellent indicator for probing cellular and subcellular microenvironments. Fluorescence lifetime imaging microscopy (FLIM) combines fluorescence lifetime measurement and microscopic techniques, offering a more powerful tool for biomedical study [8].

There are two unique advantages of FLIM compared to intensity-based fluorescence microscopy:

(a) FLIM is more quantitative.

The fluorescence lifetime is an intrinsic property of fluorophores and is less susceptive to experimental artefacts, including excitation power, optical path, fluorophore concentration, and detector gain. It facilitates the differentiation of fluorescent markers that have overlapped emission spectra. In addition, the fluorescence lifetime enables the discrimination of fluorescent signals from background signals, such as autofluorescence from cells or tissues and surrounding light, resulting in a more efficient detection of fluorescent markers and a higher image quality. The fluorescence lifetime also offers a reliable way to quantitatively measure the Förster Resonance Energy Transfer (FRET) [9, 10]. FRET is an important tool for analysing molecular dynamics and interactions. However, due to the experimental artefacts and uncertainty, the intensity-based FRET is difficult to quantify. The FRET measurement can be easily affected by laser excitation power, spectral crosstalk, and photo-bleaching. Additionally, the concentrations of both donor and acceptor fluorophores are often unknown in living cells or tissues. In contrast, FLIM offers a more accurate and robust way to analyze FRET by monitoring the lifetime change of donor and acceptor fluorophores.

(b) FLIM provides more information.

The fluorescence lifetime provides much information on fluorophores' local environment. By choosing fluorescent dyes with different molecular structures, the fluorescence lifetime can be a function of various biophysical parameters, including ion concentration (Cl^-, Cu^{2+}, K^+, Mg^{2+}, O_2, PO_4^{3-}), temperature, pH, viscosity, and so on [11]. FLIM, therefore, becomes an arsenal to sense various parameters and biological processes. The FLIM-FRET techniques can reveal protein-protein interactions and protein conformational changes without approaching molecular levels [9].

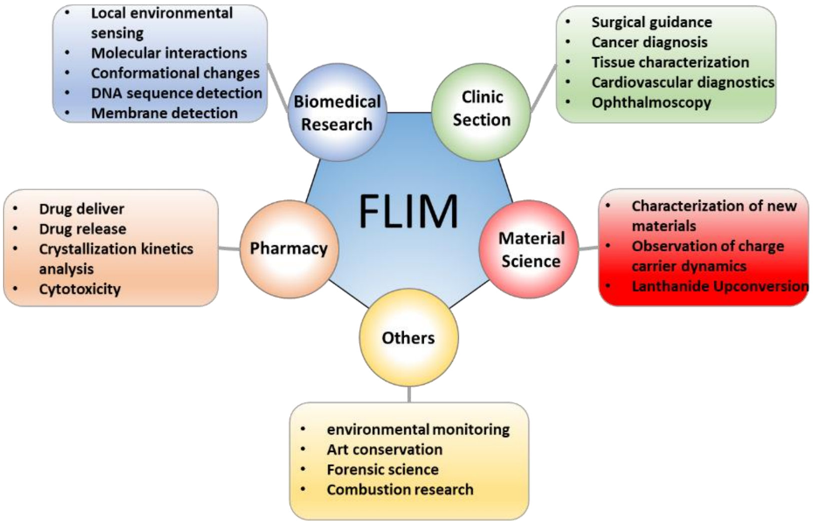


Fig. 1. FLIM applications in various disciplines

As illustrated in Fig. 1, FLIM has been successfully employed across various disciplines, showcasing its versatility. Apart from its significance in fundamental biomedical research, FLIM also holds crucial applications in clinics and pharmacies. In clinical settings, FLIM provides a feasible method for characterizing tissues using both endogenous and exogenous fluorescence. Of particular interest are label-free FLIM images that utilize autofluorescence from endogenous fluorophores, which have gained significant attention for clinical applications. Autofluorescence-based FLIM enables fast detection of metabolic changes in tissues, especially in precancerous lesions, aiding in diagnosis and surgical guidance. This technique has found applications in detecting skin, oral, cardiovascular and brain diseases, and cancers [12-15]. In addition, retinal autofluorescence imaging is a critical diagnostic tool in ophthalmology. Fluorescence lifetime imaging ophthalmoscopy has emerged as an important tool for investigating the pathophysiology of various macular and retinal diseases [16, 17]. An advanced autofluorescence-based FLIM tool, FLImBrush, has been developed for real-time analysis of FLIM images and surgical guidance in the operating room [18]. FLIM also holds significance in pharmaceutical research, where it can monitor drug delivery, investigate drug uptake kinetics, and track the drug release process [19-21]. In non-biomedical fields, FLIM has found novel applications in recent years. In material science, FLIM can characterize new materials, such as semiconductors, photovoltaics, quantum dots, and OLED [22-25]. Time-

Resolved Photoluminescence, a technique similar to FLIM, has been used for directly observing charge carrier dynamics and up-conversion photoluminescence of lanthanide complexes [26, 27]. Moreover, FLIM plays a crucial role in environmental monitoring, forensic science, combustion research, and art conservation [28-31]. With the continual emergence of new FLIM applications, its significance is expected to grow across various research fields.

FLIM techniques have achieved tremendous success, yet they still encounter several limitations and challenges. One of the most significant and persistent hurdles is the slow speed of FLIM, stemming from two primary factors: the long acquisition time and the complex and time-consuming data processing. Several factors lead to a long acquisition time:

Firstly, the Time-Correlated Single Photon Counting (TCSPC) technique is the gold standard for fluorescence lifetime measurement due to its best photon efficiency, temporal resolution, and SNR [32, 33]. However, the photon detection rate in standard TCPSC systems should keep at a low level (less than 5% laser repetition rate) to avoid pile-up effects.

Secondly, FLIM necessitates a high photon budget, which implies that it takes a considerable amount of time to gather adequate photons for lifetime analysis. This becomes particularly challenging during live-cell experiments, where detecting a sufficient number of photons within a limited time is often difficult due to various factors. For instance, the low concentration of fluorophores or unfavorable optical properties like unsuitable excitation/emission, low quantum yield, and photosensitivity can pose significant challenges. Additionally, for some exogenous fluorophores such as fluorescent proteins, low transfer, transcription, or translation efficiency of genes can result in weak signals.

Thirdly, High-resolution FLIM images are necessary for many applications, but achieving them using the standard method requires a smaller focal point and more scanning steps in laser scanning FLIM systems. Due to the above reasons, current FLIM systems are only suitable for measuring steady samples mounted on a microscopic stage. The extended measurement time inevitably causes photon damage and photon bleaching.

The complex and time-consuming data processing involved in FLIM is also a significant challenge. Since FLIM is an indirect imaging technique, the raw data should be processed to obtain the fluorescence lifetime. In TCSPC-based FLIM, the fluorescence lifetimes are calculated by fitting decay histograms with pre-defined models such as single- and bi-exponential decay models. The fitting procedure depends on iterative optimization algorithms such as the least square fitting (LSF) method, maximum likelihood estimation (MLE) method, and Bayes method [34-36]. However, these methods are computationally intensive, time-consuming, and only suitable for offline analysis. They also require prior knowledge to set proper initial conditions, and specialized software is needed to implement these algorithms. While some fitting-free methods like the phasor approach and the center-of-mass method (CMM) have been proposed for fast FLIM analysis [37-40], all existing FLIM algorithms are noise-sensitive and require high SNR for accurate analysis.

The slow working speed of FLIM severely limits its application range. Current FLIM systems are unsuitable for time-constrained applications, such as investigating transient fluorescence phenomena, quickly assessing dim fluorophores, or performing high throughput fluorescence lifetime sensing. Furthermore, the complex data analysis involved makes it challenging to design portable FLIM devices for real-time measurements. The research in this thesis aims to address the above challenges using data-driven deep learning (DL)-based computational approach, without any modifications to the hardware implementation of FLIM systems. The specific aims of this thesis are as follows:

1. To develop DL algorithms that can perform fast and robust FLIM analysis. These algorithms will surpass all existing FLIM analysis methods in terms of accuracy, speed of computation, and noise tolerance.

2. To develop DL algorithms with novel functionalities, such as significantly reducing the measurement time in laser scanning FLIM systems and decreasing the large photon budget required for existing FLIM algorithms.

3. To design a time-resolved flow cytometry system that integrates an advanced single-photon avalanche diode (SPAD) array sensor and DL algorithm. The system will enable high throughput and fluorescence lifetime sensing.

1.2 Contributions to knowledge

The contributions to knowledge resulting from this thesis can be summarized as follows:

1. We proposed a novel DL method, the pixel-wise DL algorithm, to address the challenge of complex FLIM data analysis. This algorithm provides unparalleled advantages, including a high-efficient architecture with fewer parameters, a shorter algorithm training time, faster analysis speed, and a strong ability to resolve multi-exponential decay models. Compared to conventional fitting methods, the proposed algorithm improves calculation speed by more than 300-fold while also significantly improving estimation accuracy for all decay parameters. Our algorithm is well-suited for various computing platforms, including embedded systems, due to its simple implementation, high computation efficiency, and low consumption of hardware resources.

2. We developed a computational framework to overcome the challenge of limited available FLIM data for developing DL FLIM techniques. Our approach can generate large-scale semi-synthetic FLIM datasets with various cellular morphologies, a sizable dynamic lifetime range, and multiple decay components. This offers a feasible solution to the problem, avoiding the need for labor-intensive and time-consuming experimental work.

3. We developed DL algorithms to extend the capability of current FLIM systems without modifying the system configurations. Specifically, we proposed a DL algorithm for improving the spatial resolutions of FLIM images. Our algorithm can obtain high-resolution (HR) fluorescence lifetime images from low-resolution (LR) images acquired from FLIM systems. It shows superior performance in reconstructing spatial information from limited pixel resolution, offering a practical approach to fast obtaining HR FLIM images.

4. The large photon budget required for quantitatively analyzing fluorescence lifetimes has been a longstanding challenge in existing FLIM measurements. To address this challenge, we proposed a DL algorithm for analyzing FLIM images with few photons. Our algorithm pushes the photon budget to a record-low level of only a few photons per pixel, enabling robust evaluation of FLIM images with high accuracy in extreme photon starve scenarios, even when the average photon-per-pixel is close to one. To the best knowledge, this is the first available method for analyzing fluorescence lifetime images in such a low photon condition. Our DL approach offers significant advantages over traditional FLIM analysis algorithms and promises broader applications in fundamental biological research.

5. We developed a time-resolved flow cytometry system aimed at fast fluorescence lifetime measurement. Our design includes a parallel light detection scheme that utilizes the advanced CMOS 192x128 single-photon avalanche diode (SPAD) array to overcome the limited photon-counting throughput typical of conventional sensors. The SPAD array operates as a point-like sensor, providing a large sensing area and more than 20,000 parallel photon-counting channels, resulting in an excellent photon-counting throughput. We also developed a field-programmable gate array (FPGA) embedded DL processor based on the pixel-wise DL algorithm for fast fluorescence lifetime analysis. The pixel-wise DL algorithm is further simplified and quantized for hardware implementation. Our processor shows the advantages in terms of high accuracy, fast calculation speed, less power consumption, and low requirement of hardware resources. Our flow cytometry system can effectively measure fluorescence lifetimes from noise-corrupted data within a single frame. Additionally, it has a small footprint and paves the way for portable time-resolved devices.

1.3 Thesis overview

This thesis has seven chapters. The first chapter is introductory, providing an overview of the research motivations and aims, the contributions to knowledge achieved in this thesis, and outlining the thesis organization. The organization of the remaining thesis is as follows:

Chapter 2 presents a literature review of the background knowledge, which is divided into two parts. The first part provides an introduction to the basic concepts of fluorescence, the measurement principles of fluorescence lifetime, FLIM systems, and a review of FLIM data analysis methods. The second part offers a comprehensive overview of the state-of-the-art DL techniques, including the fundamental concepts, cutting-edge DL architectures, and the optimization and regularization methods for network training. The final section discusses the recent advances of DL in microscopic imaging and FLIM.

Chapter 3 presents the pixel-wise DL algorithm for fast FLIM analysis. The Monte-Carlo simulation method is first used to generate synthetic training data with wide lifetime and SNR dynamic ranges. Then, the details of neural network design and training are presented. The performance of the trained network in resolving multi-decay models is investigated using synthetic testing datasets. The pixel-wise DL algorithm is compared with other methods to demonstrate its unique advantages. The proposed algorithm is further employed to analyze two-photon FLIM images of functionalized gold nanoprobes in Hek293 and human prostate cancer cells to validate its superior performance. The main content in Chapter 3 is adapted from the paper published online:

- **D. Xiao**, Y. Chen, and D. D.-U. Li, "One-Dimensional Deep Learning Architecture for Fast Fluorescence Lifetime Imaging," *IEEE Journal of Selected Topics in Quantum Electronics,* vol. 27, no. 4, pp. 1-10, 2021, doi: 10.1109/jstqe.2021.3049349.

Chapter 4 introduces the spatial resolution improved FLIM net (SRI-FLIMnet), a DL algorithm that enhances the spatial resolution of fluorescence lifetime images by generating high-resolution (HR) images from low-resolution (LR) images. Semi-synthetic FLIM training datasets are created using a computational method that combines the mathematical model of the forward TCSPC process and Human Protein Atlas (HPA) datasets. These FLIM datasets possess distinct spatial and temporal features. A degrading model is developed to describe the transformation from HR to LR fluorescence lifetime images in FLIM systems. The design of SRI-FLIMnet is presented in detail, and its performance is evaluated using synthetic datasets and experimental data of bacterial-infected mouse raw macrophage cells, with a focus on the reconstruction of fluorescence lifetime and spatial information. The main content in Chapter 4 is adapted from the paper published online:

- **D. Xiao**, Z. Zang, W. Xie, N. Sapermsap, Y. Chen, and D. D. Uei Li, "Spatial resolution improved fluorescence lifetime imaging via deep learning," *Opt Express,* vol. 30, no. 7, pp. 11479-11494, Mar 28 2022, doi: 10.1364/OE.451215.

Chapter 5 presents the FPFLI DL algorithm, which is designed to address the challenge of analyzing fluorescence lifetime images under extremely low light conditions. When the photons are few in a decay histogram, the lifetime estimation becomes an ill-posed problem. The FPFLI algorithm adopts a hybrid DL technique to evaluate local lifetime and fuse the information of local lifetime and intensity images. This chapter provides a detailed description of the algorithm's design and implementation. The performance of the proposed algorithm is quantitatively evaluated using synthetic FLIM data. The proposed algorithm is validated with experimental data, including fluorophore-tagged beads and FLIM-FERT imaging of HeLa cells. The main content in Chapter 5 is adapted from the paper in preparation:

- **D. Xiao**, Y. Chen, and D. D. U. Li, "Deep learning enhanced fluorescence lifetime imaging with a few photons," bioRxiv, https://doi.org/10.1101/2023.04.06.534322.

Chapter 6 focuses on the application of DL algorithms to dynamic fluorescence lifetime measurements. As an example, this chapter presents the development of a time-resolved flow cytometry system that can measure the fluorescence lifetime of fast-moving particles/cells. This chapter begins by introducing flow cytometry and briefly reviewing cutting-edge time-resolved flow cytometry systems. We then summarize the challenges for fast lifetime measurement and present the design of the flow cytometry system to illustrate how we address current challenges. To improve the photon-counting throughput and provide a large sensing area, we configured the advanced CMOS 192x128 SPAD array as a point-like sensor in a parallel light detection scheme. We quantized the pixel-wise DL algorithm using a mixed-precision quantization scheme and developed a DL processor that was quantitatively evaluated on an FPGA board for fast lifetime analysis. We present the calibration of the whole system and evaluate the performance of lifetime measurements with fluorophore-tagged microspheres. The results demonstrate that the time-resolved flow cytometry system can accurately measure the fluorescence lifetimes of noise-corrupted fluorescence signals from fast-moving particles. The main content in Chapter 6 is adapted from the paper published online:

- **D. Xiao**, Z. Zang, N. Sapermsap, Q. Wang, W. Xie, Y. Chen, and D. D. U. Li, "Dynamic fluorescence lifetime sensing with CMOS single-photon avalanche diode arrays and deep learning processors," *Biomed Opt Express,* vol. 12, no. 6, pp. 3450-3462, Jun 1 2021, doi: 10.1364/BOE.425663.

Chapter 7, the final chapter, summarizes the key conclusions in this thesis, discusses the limitations and future work to be improved, and outlooks future research directions.

Chapter 2

literature review

2.1 The fundamentals of fluorescence

Fluorescence was first described by British scientist Sir George Gabriel Stokes in 1852 [41]. Since then, fluorescence has become a crucial tool in biomedical imaging and significantly advanced the fields of biology, pharmacy, chemistry, and physics [8, 11]. Fluorescence is a phenomenon of light emission from a fluorophore (some atoms and molecules) that is excited by extra electromagnetic energy. The Jablonski energy diagram, named after Alexander Jablonski, provides a visual representation of the fluorescence process of a fluorophore. As shown in Fig.2.1, when a fluorophore absorbs photons (the quantum of electromagnetic energy), the weakly bound electrons in the fluorophore are excited from the ground state S_0 to higher electronically excited singlet state S_n, without undergoing spin conversion, in femtoseconds (10^{-15} seconds). The excited molecule is in a metastable state, and after a short time of picoseconds (10^{-12} seconds), the fluorophore relaxes to the lowest vibrational energy level in the first electronically excited state, S_1. From there, the molecule returns to the ground state S_0, releasing photons in nanoseconds (10^{-9} seconds). During this process, the fluorophore can return to the ground state via two possible pathways. One pathway is radiative decay, which is responsible for fluorescence emission. The other way is non-radiative decay, including internal conversion and intersystem crossing from the singlet state to the triplet state. Due to energy dissipation in the processes such as vibration relaxation and internal conversion, fluorescence typically has lower energy or longer wavelength than absorption photons. This phenomenon is known as the Stokes shift and is advantageous for separating fluorescence from excitation light using optical filters.

Meanwhile, electrons transitioning to the triplet state can release phosphorescence emission upon returning to the ground state. Compared to fluorescence, phosphorescence has a much longer wavelength and lifetime. Both fluorescence and phosphorescence are types of photoluminescence phenomena.

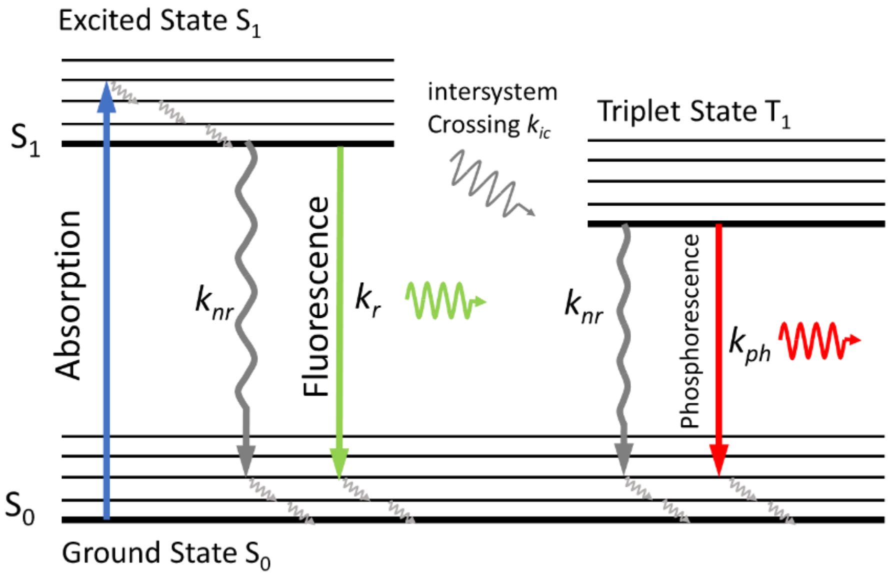


Fig. 2.1. Simplified Jablonski energy diagram to illustrate the molecular fluorescence process. S_0 and S_1 represent the ground and first excited electronic states, and the horizontal lines represent different vibrational states of the fluorophore. The blue vertical line represents the exciting light, while the green and yellow lines represent the fluorescence and phosphorescence emission. The grey vertical wavy lines refer to the non-radiative decay.

The fluorescence lifetime τ is the average time a molecule stays at the excited state before returning to the ground state. Both non-radiative and radiative decay rates contribute to the value of τ, and it is described as the inverse of the sum of all decay rate constants:

$$\tau = \frac{1}{k_{nr} + k_r}, \tag{2.1}$$

where k_{nr} and k_r are the non-radiative and radiative rate constants, respectively. k_{nr} is the sum of the internal conversion rate k_{ic}, and the intersystem crossing rate k_{isc}, so that $k_{nr} = k_{ic} + k_{isc}$.

Assuming a group of fluorophores with population N_0 are excited at time $t = 0$, the fluorescence decay follows the stochastic exponential decay distribution as the following equation:

$$dN/dt = -(k_{nr} + k_r)N(t), \tag{2.2}$$

where $N(t)$ is the population of excited fluorophores at time t. By integrating this equation, $N(t)$ expresses as $N(t) = N_0 e^{-t/\tau}$. Since fluorescence intensity $I(t)$ is proportional to $N(t)$, it also writes as:

$$I(t) = I_0 e^{-t/\tau}, \tag{2.3}$$

where I_0 is the initial fluorescence intensity at $t = 0$. It shows that the fluorescence decay follows the exponential decay law. If there are several fluorophores both contributing to the fluorescence emission, the total measured intensity is the sum of all exponential decays:

$$I(t) = \sum_{i=1}^{n} A_i \, e^{-t/\tau_i}, \tag{2.4}$$

where A_i and t_i are the intensity and lifetime for i^{th} fluorophores, respectively.

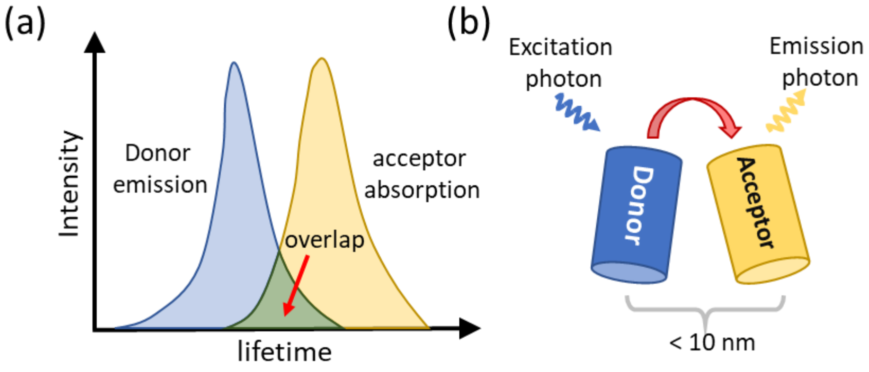


Fig. 2.2. Conditions of the occurrence of FRET. (a) The donor emission and acceptor absorption spectra should overlap. (b) The donor and acceptor should be within 10 nm and have correct orientations.

Förster resonance energy transfer (FRET) is a phenomenon in which a fluorophore in the excited state (donor) transfers energy to another nearby fluorophore in the ground state (acceptor) through a non-radiative process. FRET is named after the German scientist Theodor

Förster, who first observed this phenomenon. As shown in Fig 2.2, there are two necessary conditions for the occurrence of FRET. First, the fluorescence emission spectra of the donor should have a spatial overlap with the fluorescence excitation spectra of the acceptor (Fig 2.2(a)). Second, the distance between two fluorophores should be close to less than 10 nm, and their orientation should match each other (Fig 2.2 (b)). Direct visualization of protein and molecular interactions within several nanometers is challenging because conventional microscopy has a physical limitation of spatial resolution of around 200 nm. On the other hand, FRET offers an indirect route to investigate protein interactions and distributions by monitoring the energy transfer between donor and acceptor.

Fluorescence lifetime provides a distinct route to measure FRET quantitatively. As mentioned above, the fluorescence lifetime of a fluorophore depends on both radiative and nonradiative processes. FRET leads to fluorescence lifetime changes in both the donor and acceptor. The energy transfer efficiency of FRET can is:

$$E_{FRET} = 1 - \frac{\tau_{DA}}{\tau_D}, \tag{2.5}$$

where τ_{DA} is the fluorescence lifetime of the donor in the presence of the acceptor and τ_D is the lifetime of the donor alone.

2.2 Measurement principles of fluorescent lifetime

The typical lifetimes of fluorophores range from picoseconds to nanoseconds, thus requiring ultrafast electronics and photon detectors to acquire lifetime images. The measurement of fluorescence lifetime can be implemented either directly in the time domain (TD) or indirectly in the frequency domain (FD) [8]. In the TD method, the fluorophore is periodically excited by a pulsed laser, and the fluorescence decay is directly recorded in the histogram. In the FD method, the fluorophore is excited by a continuous modulated laser. The lifetime is calculated from the phase and amplitude difference between the exciting signal and fluorescence. Although both TD and FD methods are mathematically equivalent and related by Fourier transform, the specific implementation for both methods is different. Each method has pros and cons and needs a trade-off in practical implementations. For example, there are ultrafast electronics for distinguishing photon arriving time at picosecond precision. However, no such electronics can discriminate the tiny spectral difference between photons in the frequency domain. The FD method generally requires less complicated hardware and simple data processing. It’s more common than the TD method in past decades. However, due to the emergence of new ultrafast lasers, single-photon detectors, and data processing systems, the TD FLIM systems have attracted increased attention recently due to their unparalleled advantages, such as high accuracy, high photon efficiency, and large dynamical temporal range.

2.2.1 Frequency-domain method

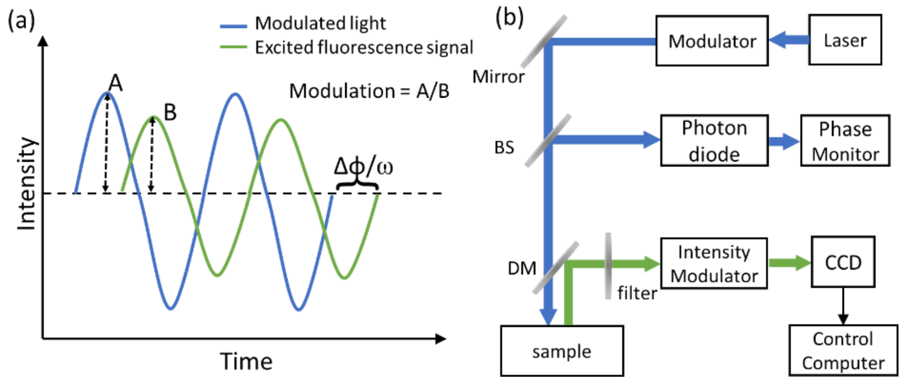


Fig. 2.3. (a) Principle of the frequency-domain method. (b) Block diagram of the frequency-domain lifetime measurement setup. BS is an abbreviation for a beam splitter, DM for the dichroic mirror, and CCD for the charge-coupled device.

The basic principle of the FD method is shown in Fig. 2.3 (a). A sinusoidal wave modulates a laser source. The sample fluorescence emission is modulated by the same frequency accordingly. However, the amplitude is demodulated, and the phase is delayed to some extent due to fluorescence decay kinetics. The fluorescence lifetime can be calculated from the demodulated factor *m=B/A* and phase angle φ using the following:

$$tan\phi = \omega\tau_{\phi}, \tag{2.6}$$

$$m = \frac{1}{\sqrt{1+\omega^2\tau_m^2}}, \tau_m = \frac{1}{\omega}\left[\frac{1}{m^2}-1\right]^{1/2}, \tag{2.7}$$

where τ_m and τ_φ are the modulation and phase lifetime for the decay curve. The experimental setup of the FD method is briefly illustrated in Fig. 2.3 (b). A radio frequency wave modulates a laser source at 1 - 100 MHz. The measurement of different lifetimes requires different modulating frequencies. Generally, Long-lived fluorophores require lower modulation frequency, whereas short-lived fluorophores require higher modulation frequency. In FD measurement, it's not easy to accurately estimate the phase angle and modulation, especially for higher frequency. Hence, FD systems need careful optimization and data analysis.

2.2.2 Time domain method

Thanks to the vast advance in photon detectors, timing electronics, and ultrafast laser technologies, time-domain FLIM measurements have become widespread with the merits of high accuracy, photon efficiency, and high SNR. The time-domain FLIM measurement techniques are divided into time-gated (TG) and time-correlated single-photon counting (TCSPC) methods [33, 42-44]. For the TG method, the arrived photons are integrated within one or several delayed time windows to estimate lifetimes. For the TCSPC method, the arrival time of all photons is recorded and accumulated in each time bin to form a histogram. Then the lifetime is recovered from the histogram.

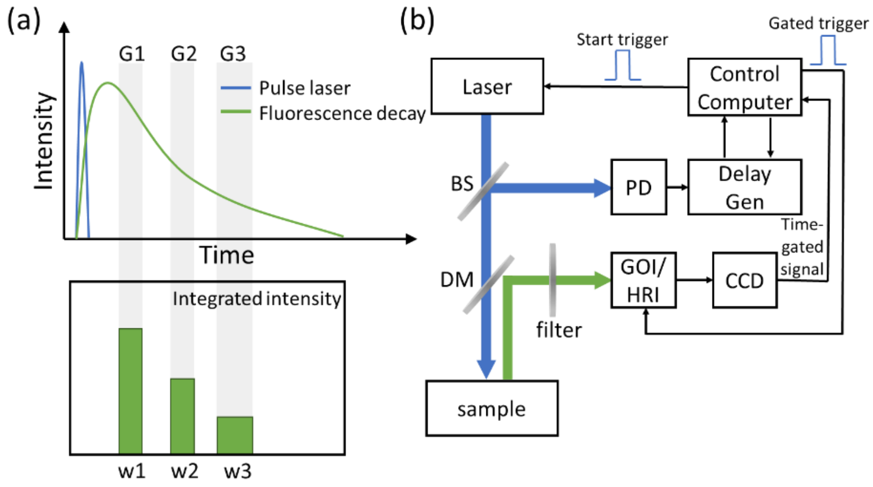


Fig. 2.4. (a) Principle of the time-gated method. As an example, three gates (G1 – G3) with different gate widths are used (w1 – w3). The photon intensity is integrated with each gate. (b) Block diagram of the time-gated measurement setup. BS is an abbreviation for beam

splitter; DM for the dichroic mirror; PD for photodiode; Delay Gen for delayer generator; CCD for the charge-coupled device; GOI for gated optical intensifier; HRI for the high-rate imager.

Fig. 2.4 conceptually illustrates the TG method. Fig. 2.4 (a) depicts its principle. After the excitation laser pulse, the emitted fluorescence photons are counted in several delayed time gates. By optimizing the delay amounts and widths of different time gates, the lifetime can be recovered by the recorded segments of the histogram in each time gate. Fig. 2.4 (b) illustrates the experiment setup. The pulse excitation single from the laser source is split into two paths by a beam-splitter. One beam excites the sample, while another is a trigger signal in the delay generator. The delay generator can control the detector with a programmable time delay. During the time-gated windows, the emitted fluorescence photons are collected and integrated by the detector. Different segments of decay curves are acquired in the measurement. The fluorescence lifetime can be reconstructed using the intensities of different segments.

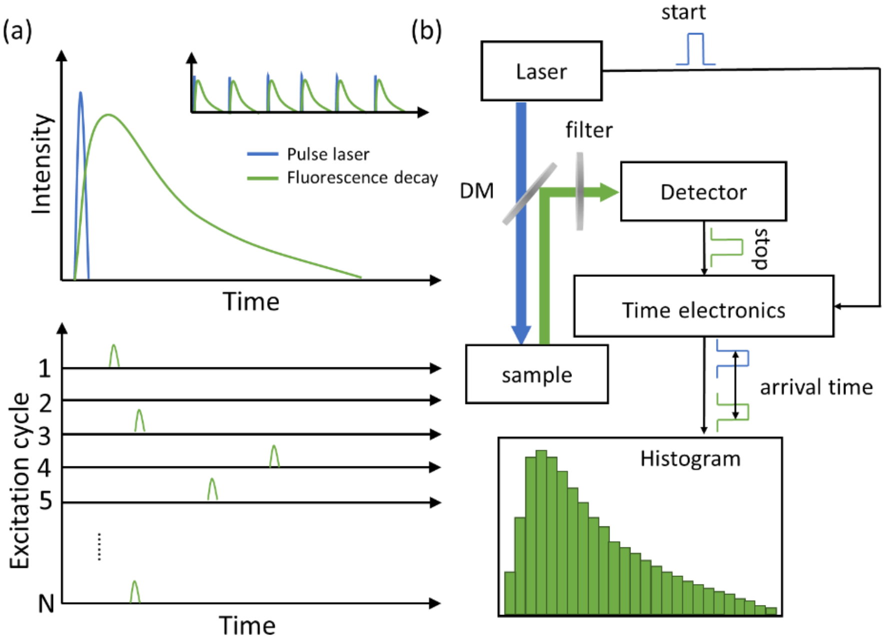


Fig. 2.5. (a) Principle of TCSPC measurement. (b) A generalized TCSPC measurement setup. The core timing electronics is TDC, which records the arriving photons with a time stamp. All the photon events are stored in a histogram.

Fig. 2.5 shows the principle of the TCSPC method. The sample is periodically excited by a laser source with a high reputation rate. The laser power is lower to avoid photodamage to the sample, leading to weak fluorescence emission. Therefore, the probability of detecting a single photon in each excitation period is much less than one, and the detector detects at most one photon each time. Fig. 2.5 (b) shows the general experimental setup for TCSPC measurement. A picosecond pulsed laser works as the light source to excite the sample. A photomultiplier (PMT) or single-photon avalanche diode (SPAD) sensor is used as a single-photon detector to collect the fluorescence signal [44]. The state-of-art timing electronics is a time-to-digital converter (TDC), which measures the arrival time of the detected photon. The working principle of TDC is as follows. The laser pulse will give a start trigger signal to TDC. When detecting a photon, the detector will send a stop signal to TDC. Hence, the TDC will record a digital code representing the photon arrival time relative to the latest start signal. After accumulating enough photon arrival events, the decay curve can be directly reconstructed from the envelope of the histogram to estimate the lifetime.

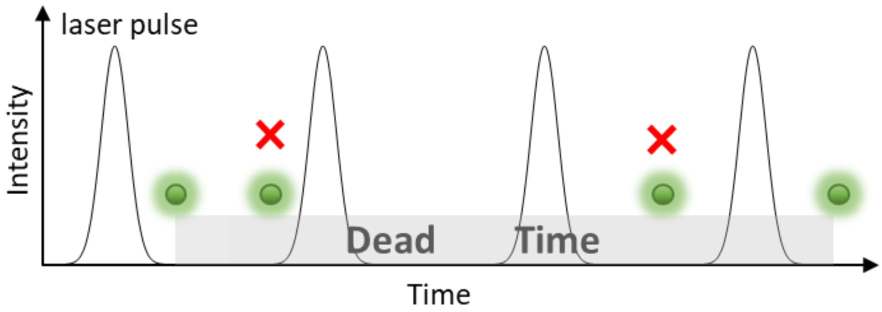


Fig. 2.6. Pile-up effect in TCSPC system. The detected photons cannot be registered within the dead time. Photons with a longer lifetime are discarded, and the measured histogram skews to a shorter lifetime.

In TCSPC, the “dead time” and “pile-up” effect are primary concerns, as illustrated in Fig. 2.6 [45, 46]. The detector and TDC in the TCSPC system are not ideal components; after detecting one photon, they require a relatively long time to recover to the “work state”, during which the system cannot detect other photons. This time interval is referred to as the dead time. In common TCSPC systems, the typical dead time ranges from 25 to 100 ns, while the interval between excitation pulses is 25 ns for a 40MHz laser. If one photon is counted and processed in an excitation cycle, the successive 3 to 4 excitation cycles cannot count other photon events. Consequently, the photon-counting rate is limited to 1% - 10% of the laser repetition rate. Failure to meet this condition results in the pile-up effect, where only the first arriving photon is counted, and later arriving photons are discarded, leading to an overrepresentation of early photons in the statistics. The histogram distorts, and the lifetime becomes shorter than the actual value. The pile-up effect limits the photon counting throughput of a TCSPC system. To alleviate the pile-up effect, advanced sensors and TDCs with a short dead time can be developed, or multi-channel systems can be developed to route detected photons to different channels.

Both TG and TCSPC methods are widely used in various FLIM systems, each with its advantages and limitations. Compared to the TCSPC FLIM system, the hardware for TG systems only requires time-grating electronics, making it simpler and more power-efficient. Since there are no TDC electronics, the TG system has a shorter dead time and is less susceptible to the pile-up effect. The frame acquisition rate is much faster than the TCSPC method accordingly. The TG method can be used for wide-field rapid lifetime imaging in combination with the rapid lifetime determination (RLD) algorithm. However, the TG method has worse time resolution than the TCSPC method, and photon efficiency is relatively low since photons are only counted in the time-gated window.

The main advantage of the TCSPC method is photon efficiency and high accuracy. All detected fluorescence signals can be processed, and the SNR of TCSPC is close to the ideal value corresponding to Poisson statistics. For this reason, TCSPC has become the standard method for FLIM. However, TCSPC also has limitations. One significant limitation is the slow acquisition time required to obtain sufficient photons for reliable data analysis, which significantly hampers the application of TCSPC in rapid lifetime imaging, such as investigating fast processes in biological samples, dynamic FRET, and high throughput FLIM screening. Another drawback of TCSPC is the extensive computational consumption, where a dedicated FLIM algorithm needs to calculate the lifetime from the measured data.

2.3 Fluorescence lifetime imaging microscopy systems

There are two microscopy modalities for FLIM: laser scanning (LS) and wide-field (WF) microscopies [46-50]. LS-FLIM systems adopt a single-point sensor to scan the sample and detect the fluorescence decay pixel-by-pixel. According to the fluorescence excitation method, LSM-FLIM can be divided into confocal LS (CLS) and multiphoton LS (MLS) FLIM systems [51-53]. On the other hand, WF-FLIM systems use parallel illumination to excite the sample. A position-sensitive sensor simultaneously detects the fluorescence signals. This section reviews the experimental configurations of both systems and discusses their advantages and disadvantages.

2.3.1 Laser scanning FLIM systems

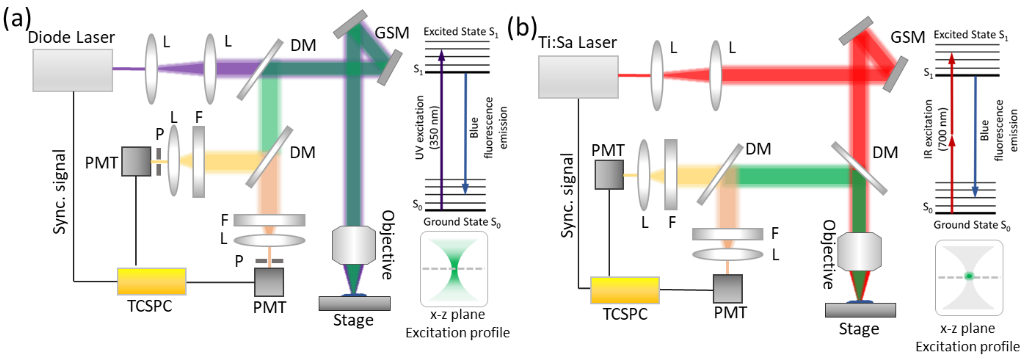


Fig. 2.7. Schematic of (a) the two-channel CLS-FLIM system and (b) the MLS-FLIM system. The inserts shown in (a) and (b) are the simplified Jablonski energy diagram of the light emission processes of the samples. The excitation profiles on the x-z plane are also shown in the inserts. Abbreviations for optical elements: DM: dichroic mirror; GSM: galvanometer scanning mirror; L: lens; F: filter; P: Pin hole.

LS-FLIM systems are widely used tools in biomedical research. Fig. 2.7 depicts the experimental setups of two-channel CLS-FLIM and MLS-FLIM systems. Both systems utilize ultrafast mode-locked laser sources for fluorescence excitation. Diode lasers are a common choice for CLS-FLIM systems due to their ultra-short pulse duration (several hundred femtoseconds), durable repetition frequency, and wide dynamic wavelength range from ultraviolet to infrared. On the other hand, Ti: sapphire lasers are popular in MLS-FLIM systems due to their tunable wavelength range from 720 to 900 nm. The scanning system involves a pair of galvanometric mirrors that precisely control the movement in the X-Y plane, while the microscopic stage and objective lens control the focal plane position in the Z direction. Photomultiplier tubes and a TCSPC card record the time-tagged photon event. As shown in Fig. 2.7 (a), In CLS-FLIM systems, the one-photon excitation principle is used, where the light direct excites the sample within the excitation volume. The fluorescence in the 3D volume returns along the same path as the excitation light, retracing the scanning mirrors before passing to the detector. This optical path is referred to as the descanning path. A pinhole is placed in front of the detector to reject the out-of-focus fluorescence and produce a thin optical section.

In contrast, MLS-FLIM systems adopt multi-photon excitation principles, including two-, three-, and four-photon excitations. Fig. 2.7 (b) shows an example of the two-photon LS-FLIM system. The fluorescence is excited by simultaneously absorbing two photons, where each photon contributes half the energy. Because the energy of a photon is inversely proportional to its wavelength, the two absorbed photons should have a wavelength twice longer than that required for one-photon excitation. The photon intensities for two-photon excitation are one million times larger than that of one-photon absorptions. Fortunately, modern pulse lasers easily meet this condition as the excitation light highly concentrates spatially and temporally. The photon density is extremely high within a pulse duration to generate significant two-photon excitation, while the ultra-short duration keeps the average light power at a very low level (less than ten milliwatts). Unlike CLS-FLIM systems with a large excitation volume, the effective excitation of MLS-FLIM systems only occurs around the focal point. Therefore, the pinhole is no longer necessary, and the optical path is more flexible. A dichroic mirror is placed after the objective lens to separate and reflect the fluorescence to detectors. Such an optical path is also called the non-descanning path. It can collect more scattered photons, require fewer optical elements, and reduce the path length.

CLS-FLIM systems have several disadvantages, which can be summarized as follows:

(a) **Photon damage**. Due to the large excitation volume, the excitation light generates significant photobleaching and phototoxicity problems, especially in live cells.

(b) **Low photon efficiency**. The excitation light is strongly absorbed by the tissues within the excitation volume, resulting in only a small portion of the excitation photons reaching the focal point to excite fluorescence. Moreover, the fluorescence photons emitted from the focal plane may scatter by tissues and get blocked by the pinhole, leading to a further decrease in photon efficiency.

(c) **Limited penetration depth**. The strong absorption of excitation light severely limits the penetration depth in thick specimens. The photon scattering from the out-of-focus plane also decreases the image SNR.

MLS-FLIM systems provide a promising solution to the issues mentioned above. In MLS-FLIM systems, tissues cannot absorb the excitation photons with a longer wavelength. The fluorescence is only generated at the focal point, resulting in minimal photobleaching and photodamage. The absence of pinhole and non-descanning optical path allows the detection of more scattered light. The excitation light can reach deeper areas by eliminating out-of-focus absorption. Additionally, a longer wavelength leads to less scattering by tissues. All these factors contribute to the excellent penetration ability of MLS-FLIM systems, making them more suitable for imaging thick tissues and in vivo specimens.

2.3.2 Wide-field FLIM systems

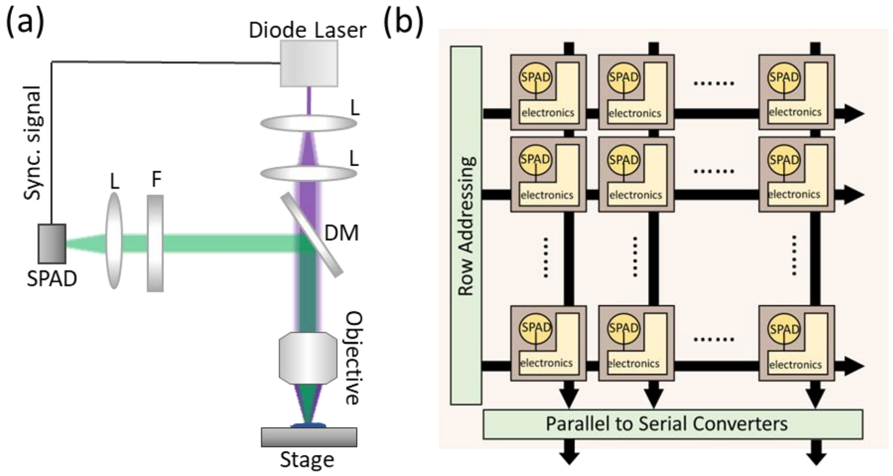


Fig. 2.8. Schematic of (a) the WF-FLIM system and (b) the SPAD array sensor

WF-FLIM systems use a parallel light field to illuminate the sample and collect the fluorescence signal from the focal plane of the objective lens. Therefore, the scanning system is no longer required, and the optical path is greatly simplified. WF-FLIM systems can adopt TG and TCSPC techniques for time-domain measurements [54-56]. Fig. 2.8 shows the schematic view of the WF-FLIM system. WF-FLIM systems require position-sensitive single-photon detectors such as PMT arrays, microchannel plates (MCPs), and single-photon avalanche diode (SPAD) arrays. SPAD arrays represent the most state-of-the-art single-photon detection [57-63]. The SPAD works in Geiger mode, which is reversely biased above the diode breakdown voltage. Unlike PMTs and MCPs, SPADs are all-solid-state photon detectors without requiring high voltage and vacuum. SPADs show excellent single-photon sensitivity and timestamping capability down to the picosecond time resolution. SPAD arrays can be integrated into a miniaturized chip by standard CMOS techniques, which are scalable and low-cost. Fig. 2.8 (b) shows the 32×32 SPAD array architecture of MEGAFRAME32 [63]. Each pixel contains a 50 ps, 10-bit time-to-digital converter for timestamping. The maximum frame rate reaches 500 kilo-frame per second. The power supply, control, and read-out circuits are shared among the columns and rows. The SPAD array can work at both TCSPC mode and photon counting modes.

The advantages of WF-FLIM systems include a fast data acquisition speed with high frame rates, less photodamage to samples, and simple implementation. The SPAD array chips can be integrated into a system-on-chip (SoC) for more complex functionalities. However, WF-FLIM systems also suffer some disadvantages. First, the lack of a pinhole in the system causes the detected signals easily contaminated by light from the out-of-focus plane and the surrounding environment. In addition, the SPAD arrays have considerable dark-count noise and relatively low sensitivity. So, the images obtained by WF-FLIM systems usually have poor SNR compared to LS-FLIM systems. Second, the axial resolution is poor as well. Current 2D SPAD arrays have a low filling factor due to the peripheral circuit occupying a large area. Meanwhile, the signals in temporal and spatial coordinates could intermix, leading to a trade-off between either spatial or temporal resolution.

2.4 Fluorescence lifetime imaging analysis

Data analysis plays a vital role in FLIM because FLIM is an indirect imaging technique. The accurate estimation of fluorescence lifetimes is a prerequisite for further analyzing cell functions. The fluorescence lifetime analysis methods can be categorized into two types: curve-fitting methods and fitting-free methods. This section reviews the commonly used methods of both types.

2.4.1 Model-based Curve-fitting methods

Curve-fitting methods are the most widely used methods to analyze fluorescence lifetimes. Curve-fitting methods aim to find model parameters that best match the measured data and the decay model, given a pre-defined decay model and measured FLIM data. A multi-exponential decay with *P* lifetime components is:

$$f(t) = \sum_{p=1}^{P} \alpha_p \, e^{-t/\tau_p}. \tag{2.8}$$

The measured fluorescence model is the convolution between *f*(*t*) and the instrument response function (IRF) *I*(*t*), which is:

$$y(t) = I(t) * y(t) + \varepsilon(t), \tag{2.9}$$

where the asterisk denotes the convolution operator, and $\varepsilon(t)$ is the noise term. Eq. (2.9) can be rewritten in discrete form:

$$y(k) = \sum_{i=1}^{k} I(k-i) f(i) + \varepsilon(t), k = 1, \cdots, N, \tag{2.10}$$

where *N* is the number of time bins. The discrete form of *f*(t) is described as:

$$f(k) = \sum_{p=1}^{P} \alpha_p \, e^{-t_s(k-1)/\tau_p}, k = 1, \cdots, N. \tag{2.11}$$

In Eq. (2.11), t_s is the time bin width. Assume the measured data is $\boldsymbol{x} = [x_1, x_2, \ldots, x_N]$ and the model estimated value is $\boldsymbol{y} = [y_1, y_2, \ldots, y_N]$ where x_i and y_i are the measured and predicted photon events in i^{th} time bin, respectively. The curve-fitting process is to solve an optimization problem by minimizing a loss function or goodness-of-fit parameter $\boldsymbol{L}(\boldsymbol{x}, \boldsymbol{y})$. Various methods have been developed for fluorescence lifetime estimation. The popular methods are summarized below:

- *Least Squares fitting (LSF) method [64]*

The LSF is a common method for curve fitting because of its easy implementation. The loss function is defined as the squared difference (L2-norm) between $\boldsymbol{x}$ and $\boldsymbol{y}$:

$$\chi_{ls}^2 = \sum_{i=1}^{N} \frac{(x_i - y_i)^2}{\sigma_i^2} = \sum_{i=1}^{N} \frac{(x_i - y_i)^2}{x_i} \tag{2.12}$$

where σ_i is the standard deviation in i^{th} time bin. In TCSPC measurement, the noise follows a Poisson distribution and σ_i can be calculated as $\sigma_i = \sqrt{x_i}$. Eq. (2.12) minimization can be solved using iterative optimization algorithms such as Levenberg-Marquardt and Trust-Region-Reflective algorithms. It is worth noting that the LS method inherently implies that the data to be fitted have a Gaussian distributed noise, which is inconsistent with TCSPC data. Therefore, the LS method can only obtain desirable results in a high photon-count condition (e.g., photon counts > 1000) where the Poissonian statistics converge to Gaussian statistics[65]. The LS method will deliver inaccurate values when the photon counts are low.

- *Least Squares fitting based on Laguerre expansion (LSF-LE) [66,67]*

The LSF-LE method expands the discrete fluorescence decay model in Eq. (2.11) onto an ordered set of orthonormal Laguerre basis functions *$b_l(k; \alpha)$* as:

$$\hat{f}(k) = \sum_{l=0}^{L-1} c_l b_l(k; \alpha), \tag{2.13}$$

Where L is the Laguerre dimension, α ($0<\alpha<1$) is the scale parameter of Laguerre basis functions, and c_l is the l^{th} expansion coefficient. The l^{th} discrete LBF is defined as:

$$b_l(k; \alpha) = \alpha^{k-l}(1-\alpha)^{1/2}(-1)^i \binom{k}{i}\binom{l}{i} \alpha^{1-i}(1-\alpha)^i, l = 0, \cdots, N. \tag{2.14}$$

The Laguerre expanded decay model in Eq. (2.10) is linearly parameterized by the expansion coefficient c_l:

$$\hat{y}(k) = \sum_{i=1}^{k} c_l I(k-i) b_l(i; \alpha) + \varepsilon(k), k = 1, \cdots, N, \tag{2.15}$$

Therefore, the optimization problem is to minimize the normalized sum of squared errors $\sum_{i=1}^{N}(\hat{y}_i - x_i) / ||\boldsymbol{x}||$. Compared with the LSF method, the LSF-LE is more robust and effective. The convergence speed is faster as well. In practice, the Laguerre parameters are often chosen empirically. To overcome this disadvantage, automated Laguerre deconvolution methods have been developed to optimize L and α during the deconvolution process [68].

- *Maximum likelihood estimation (MLE) [69,70]*

Before discussing MLE, we briefly review the basic concept of the likelihood function. Given a set of observed data ***t*** and a set of model parameters ***θ***, Bayesian Theorem tells us:

$$P(\boldsymbol{\theta}|\boldsymbol{t}) = \frac{P(\boldsymbol{t}|\boldsymbol{\theta})P(\boldsymbol{\theta})}{P(\boldsymbol{t})}. \tag{2.16}$$

In Eq. (2.16), $P(\boldsymbol{\theta}|\boldsymbol{t})$ is the posterior distribution, meaning the probability of the model parameters given the measured data. $P(\boldsymbol{t}|\boldsymbol{\theta})$ is referred to as the likelihood, meaning the probability that each possible parameter produces the observed data. $P(\boldsymbol{\theta})$ and $P(\boldsymbol{t})$ are referred to as the prior distribution and the evidence, respectively. Treating the observed data and the prior distribution are fixed, it is known that $P(\boldsymbol{\theta}|\boldsymbol{t}) \propto P(\boldsymbol{t}|\boldsymbol{\theta})$. The MLE calculates the model parameters $\boldsymbol{\theta}$ by directly maximizing the likelihood $P(\boldsymbol{t}|\boldsymbol{\theta})$. For FLIM analysis, MLE assumes x_i follows a Poison distribution. Given a corresponding mean value μ_i, the probability of observing x_i is:

$$P(x_i|\mu_i) = \frac{\mu_i^{x_i}}{x_i!}e^{-\mu_i}, \tag{2.17}$$

The principle of MLE is to maximize the likelihood function:

$$\boldsymbol{L}(x,y) = \prod_{i=1}^{N} \boldsymbol{P}(x_i|y_i), \tag{2.18}$$

For the fitting procedure, it is convenient to apply the logarithm to the likelihood function and normalize it by dividing the maximum possible likelihood $\boldsymbol{L}$ ($\boldsymbol{x}$ | $\boldsymbol{x}$), then minimize the square of the logarithmic and normalized likelihood function χ^2_{mle}:

$$\chi^2_{mle} = -2ln\left(\frac{L(\boldsymbol{x}|\boldsymbol{y})}{L(\boldsymbol{x}|\boldsymbol{x})}\right) = 2\sum_{i=1}^{M} y_i - x_i - 2\sum_{i=1,x_i\neq 0}^{M} x_i \, ln(y_i/x_i). \tag{2.19}$$

MLE has a stable performance over a wide intensity range, especially for low photon-count conditions. It can deliver accurate values even if the photons are only several hundred. However, one significant drawback of MLE is the long computation time.

- *Bayesian approach (BA) [71,72]*

BA is closely related to MLE. Its key idea is to estimate the model parameter using the prior knowledge to describe $P(\boldsymbol{\theta})$. To apply BA for analyzing FLIM data, the likelihood of a photon being detected at a particular time bin should be developed. For simplification, the mono-exponential decay model is considered. Assume the excitation period *T* has *N* time bins. Each bin has an identical bin width h. b_i represents i^{th} bin, containing photons with arrival time, $(i-1)h \leq t \leq ih$. The number of total photon events is *C*, and the photon number in i^{th} time bin is C_i. The likelihood function is:

$$P(t|\boldsymbol{\theta}) = \prod_{i=1}^{N} [\alpha p_D(t \in b_i|\alpha, \tau) * IRF(t) + (1-\alpha)/T]^{C_i}, \tag{2.20}$$

where $\boldsymbol{\theta} = (\alpha, \tau)$ is the model parameters, α is the fraction of the fluorescence signal, and $p_D \propto e^{-t/\tau}$ is the pure fluorescence decay. The second term in the product formula denotes the probability of noise photon, which follows a uniform distribution within [0, *T*]. The posterior parameter distribution $P(\boldsymbol{\theta}|t)$ is given by:

$$P(\boldsymbol{\theta}|t) = \frac{P(t|\boldsymbol{\theta})P(\boldsymbol{\theta})}{\int P(t|\boldsymbol{\theta}')P(\boldsymbol{\theta}')d\boldsymbol{\theta}'}. \tag{2.21}$$

The prior distribution $P(\boldsymbol{\theta})$ can be determined from maximum entropy [71]. Then an optimization algorithm is used to maximize the posterior distribution $P(\boldsymbol{\theta}|t)$ to obtain the estimated parameters. It can also evaluate the likelihood function by searching parameter space on a grid of uniform or using the stochastic search methods like the Markov chain Monte Carlo (MCMC) method to build $P(\boldsymbol{\theta})$ [72].

2.4.2 Model-free none-fitting methods

- *Rapid Lifetime Determination (RLD) [73-76]*

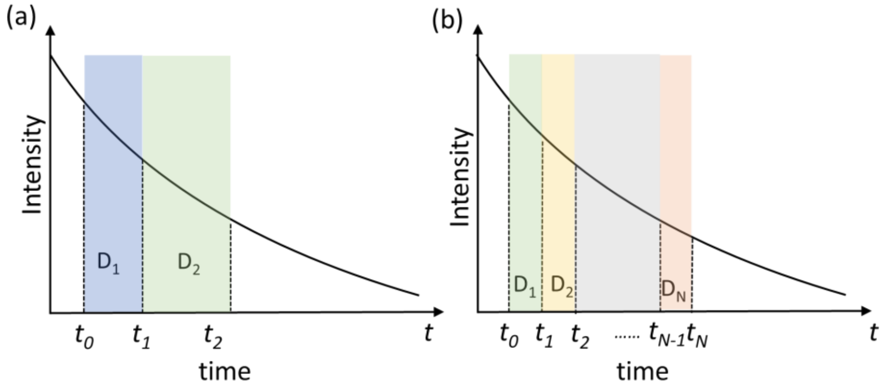


Fig. 2.9. Schematics of (a) two-gate and (b) multi-gate rapid lifetime determination (RLD) algorithms.

The RLD is a popular fitting-free fluorescence lifetime estimation method for TG-FLIM systems. Fig. 2.9 (a) shows the basic two-gate scheme. The lifetime is calculated by:

$$\tau_{RLD\text{-}2} = \frac{t_2 - t_1}{In\frac{N_{R1}}{N_{R2}}}, \tag{2.22}$$

where N_{R1} and N_{R2} are the total photon counts in gates D_1 and $D_{2,}$ respectively. A two-gate scheme is sensitive to noise. To improve the resolvability of the fluorescence decay, the two-gate scheme can also extend to an arbitrary *N* gates (RLD-N) scheme as shown in Fig. 2.9 (b):

$$\tau_{RLD\text{-}N} = \frac{\left[\left(\sum_{k=1}^{N} t_k\right)^2 - N\sum_{k=0}^{N} t_k^2\right]}{\left\{N\sum_{k=1}^{N}[t_k In(N_k)] - \sum_{k=1}^{N} t_{k\sum_{k=1}^{N} In(N_k)}\right\}}, \tag{2.23}$$

where N_k is the photon counts in the kth time bin ($t_k < t < t_{k+1}$), $j = 0, \ldots, N\text{-}1$. Although this method is straightforward, it’s not easy to implement on hardware due to the equation's natural logarithm and floating-point division.

- *Phasor analysis (PA) [37, 38]*

The PA provides a graphical representation of fluorescence decay. The data representation is changed from a classical delay histogram to a phasor plot by performing Fourier transform on the decay curve. The real and imaginary parts of the *Fourier* transform of fluorescence decays are the two coordinates of the phasor plot, respectively. Then the fluorescence collected in each pixel in the image is transformed into a single point in the phasor plot. The *g* and *s* corresponding to a given decay *I*(*t*) in the phasor plot are provided by:

$$\begin{cases} g_{i,j}(\omega) = \int_0^\infty I_{i,j}\,(t)\, cos(\,n\omega t)dt/\int_0^\infty I_{i,j}\,(t)dt \\ s_{i,j}(\omega) = \int_0^\infty I_{i,j}\,(t)\, sin(\,n\omega t)dt/\int_0^\infty I_{i,j}\,(t)dt \end{cases}, \tag{2.24}$$

where indices *i* and *j* represent the pixel position of the image and $g_{i,j}(\omega)$, and $s_{i,j}(\omega)$ are the *x* and *y* coordinates, respectively. ω is the laser repetition angular frequency, and *n* is harmonic. If the decay is a single exponential decay $I\ (t)\ =Ae^{-t/\tau}$, it's obtained that:

$$\begin{cases} g_{i,j}(\omega) = \dfrac{1}{1+(\omega\tau)^2} \\ s_{i,j}(\omega) = \dfrac{\omega\tau}{1+(\omega\tau)^2} \end{cases} \Rightarrow s_{i,j}{}^2(\omega) + (g_{i,j}(\omega) - \frac{1}{2})^2 = \frac{1}{4}, \tag{2.25}$$

Eq. (2.25) implies that all single exponential components are represented by a semicircle of center (1/2, 0) and radius 1/2 in the phasor plot. A critical property of PA is that the decay curves with different lifetimes locate in a unique position in the phasor plot. In other words, position and lifetime mapping have a reciprocal property, thus providing a mechanism for basic segmentation. PM delivers a global view of fluorescence decay at each pixel. Data analysis is done by observing the clustering of pixels, which is much faster since it is not based on calculations. The effect of the IRF on fluorescence decay can be considered as a rotation of the phasor over a fixed angle. The angle can be obtained from a reference measurement. A phasor corresponding to a shorter lifetime τ is closed to (1, 0) (where $\tau = 0$), while a phasor corresponding to a longer lifetime is closed to (0, 0) (where $\tau = \infty$). If the phasor falls on the semicircle, it can be associated with a single lifetime value. If the phasor is not on the semicircle, the corresponding fluorescence has a complex decay. For a fluorescence decay curve superposed by two decay curves with different lifetimes, all possible weighting of the two lifetimes gives a phasor distribution along a straight line. For a fluorescence decay curve superposed by three decay curves with different lifetimes, all possible weighting of the three lifetimes gives a phasor distribution in a triangle where the vertices correspond to a pure lifetime. For FRET, the phasor has s specific curved trajectories.

- *Center-of-mass method (CMM) [39, 40, 77, 78]*

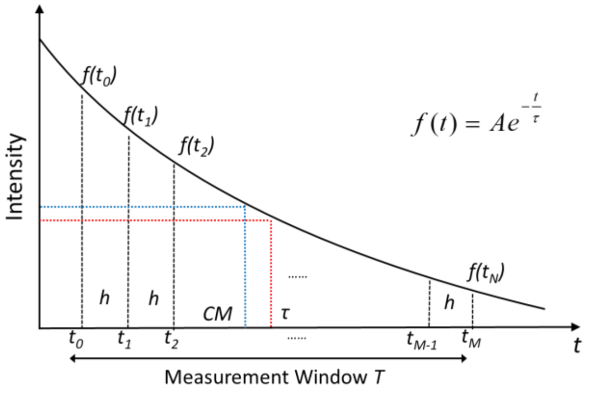


Fig. 2.10. Schematic of the single-exponential CMM. The measurement window has *M* equal bins. h is the width of the time bin. CM is the center-of-mass value, and τ is the theoretical lifetime.

The main idea of CMM is to treat the center of mass as the lifetime and compensate for the deviation. As shown in Fig. 2.10, for single-exponential decays $f(t) = Ae^{-t/\tau}$, the centre of mass is calculated as:

$$\tau_{\mathrm{CM}} = \frac{\int_0^T tf(t)dt}{\int_0^T f(t)dt} = \tau - \frac{T\,exp(-\frac{T}{\tau})}{1 - exp(-\frac{T}{\tau})}, \tag{2.26}$$

where τ is the actual lifetime and T is the measurement window. If $T \gg \tau$, the last term of Eq. (2.26) can be neglected, and the center of mass can represent τ. However, when T is comparable with τ, the deviation between τ_{CM} and τ should be calibrated. Eq. (2.26) can be written in the discrete form:

$$\tau_{CM} = \left(\frac{\sum_{i=1}^{M} i \cdot N_i}{N_C} + \frac{1}{2}\right) \cdot h, \tag{2.27}$$

where M represents the number of time bins in the measurement window T, N_i is the photon counts in i^{th} time bin, Nc is the total number of photons, and h is the time bin's width. From Eq. (2.27), it is seen that CMM is quite hardware-friendly because it only requires an accumulator to record the total arrival time of the photons (iN_i) and a counter to count the total photon number (N_c). To compensate for the deviation, a look-up table can be used to map normalized τ_{CM} to τ:

$$\tau = \Omega\left[\frac{1}{M}\left(\frac{\sum_{i=1}^{M} i \cdot N_i}{N_C} + \frac{1}{2}\right)\right] \cdot Mh, \tag{2.28}$$

where Ω is the look-up table. In Eq. (2.28), the background noise can be subtracted by placing a second background measurement window within the pre-pulse zone to sample the background signal level. The calibrated equation reads as follows:

$$\tau = \Omega\left[\frac{1}{M}\left(\frac{\sum_{i=1}^{M} i \cdot N_i - \sum\left(\frac{N_b}{S}\right)}{N_C - \left(\frac{N_b}{S}\right)} + \frac{1}{2}\right)\right] \cdot Mh, \tag{2.29}$$

where N_b and S represent the total number of photons and the number of bins, respectively in the background sampling measurement window.

- *Integral Extraction Method (IEM) [79,80]*

For a single exponential decay $f(t) = Ae^{-t/\tau}$, the fluorescence lifetime can be calculated from Eq. (2.13), where N_i is the photon counts in i^{th} time bin and h is the width of the time bin. $C_i = [1/3, 4/3, 2/3, \dots, 4/3, 1/3]$ is the coefficient of Simpson's integration rule. The coefficient can also be replaced with the coefficient of Romberg's integration rule $C_i' = [1/3, 4/3, 2/3, \dots, 4/3, 1/3]$.

$$\tau_{\text{IEM}} = \frac{h\sum_{i=0}^{M-1}(C_i N_i)}{N_0 - N_{M-1}}, \tag{2.30}$$

From the viewpoint of hardware implementation, IEM is more friendly than traditional methods like RLD since it only requires an accumulator and the denominator. This method can be further simplified by using C_i' and expresses as:

$$\tau_{\text{IEM}} \approx \left(\frac{N_c - \frac{N_0 + N_{M-1}}{2}}{N_0 - N_{M-1}}\right) \cdot h, \tag{2.31}$$

where N_c is the total number of photons. The significant advantage of IEM over RLD is much less affected by uncorrelated background noise. However, it needs to carefully choose the position and time bin width for better-optimized calculation.

2.5 The basic concepts of deep learning

2.5.1 Artificial intelligence, machine learning, and deep learning

In recent decades, Deep learning (DL) has experienced rapid growth and has had a profound impact on human society. DL has achieved great success in many fields and has brought about a revolution in industries, economy, entertainment, and healthcare. DL is a subbranch of machine learning (ML), which, in turn, is a subfield of artificial intelligence (AI). This section provides a brief overview of their fundamental concepts and relationships and explains why DL is such a powerful and successful approach to addressing various problems.

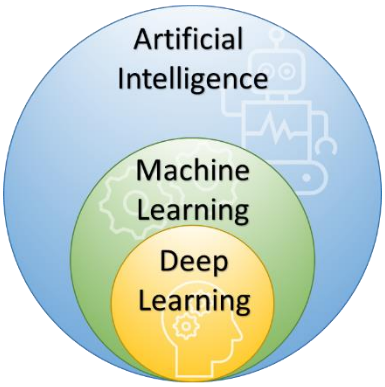


Fig. 2.11. The relationship between artificial intelligence, machine learning, and deep learning.

Fig.2.11 illustrates the relationship between AI, ML, and DL. “Intelligence” has become a buzzword in modern life, but its definition remains unclear and ambiguous as we still do not fully understand how our brain works and generates various functions such as consciousness, emotion, and memory [81]. Alan Turing, the father of computing, proposed the famous Turing Test to determine whether a machine exhibits intelligence: “If a machine engages with a human in a conversation without getting in touch with each other, it can be regarded intelligent when the human cannot detect it is a machine for a considerable time [82].” AI, which studies machines or systems with human-like behaviors and intelligence, was born out of the Dartmouth Summer Research Project on Artificial Intelligence in 1956 [83]. AI research includes the simulation of human perceptual, learning, and cognitive abilities.

ML, a crucial subfield of AI, comprises a family of algorithms that can learn general patterns from given data and make predictions on new data [84]. ML greatly contributes to the development of AI and is typically model-driven, focusing on algorithm design. The data rely heavily on manual feature engineering, which includes feature extraction, selection, and transformation. ML algorithms then make predictions based on these engineered features. Therefore, ML can be considered "shallow learning". Feature engineering depends on human experience and plays a critical role in ML, which is time-consuming and labor-intensive. Many ML problems become feature engineering problems, raising questions about what constitutes a good feature representation and how to obtain it.

The advent of DL offers a feasible way to address these questions [85]. DL adopts a "deep" algorithm to automatically learn the feature representation of data, avoiding the cumbersome feature engineering process. The "deep" refers to the algorithm's ability to obtain high-level abstract features through multiple nonlinear transformations of low-level features. Unlike ML, DL is a data-driven approach that solves problems in an end-to-end learning manner. However, DL faces a critical problem known as the Credit Assignment Problem (CAP), which is the contribution of different components (or their parameters) to the final output in a system [86]. Deep neural networks (DNNs) are the mainstream in DL because they can effectively tackle CAPs using the back-propagation training method.

Since the 2010s, DNNs have become prevalent and have outshone conventional ML algorithms, mainly due to three key factors:

(a) the availability of powerful and cheap graphics processing unit (GPU) cards, which makes training DNNs possible;

(b) the explosive growth of labeled datasets with the dawn of the internet era, which enables the training of very large DNNs with billions of parameters without overfitting; and

(c) the outstanding success of DL, which has attracted enormous commercial interest. Various technology giants such as Google, Facebook, Amazon, and Baidu are devoting significant resources to this area and developing high-quality open-source DL toolkits such as TensorFlow [87], PyTorch [88], MXNet [89], and PaddlePaddle [90]. These toolkits considerably lower the requirement for developing DL algorithms and attract more researchers to DL areas.

2.5.2 Artificial neural network

Artificial neural networks (ANNs) are mathematical models inspired by the structure and function of the nervous system in human brains. ANNs are also commonly referred to as neural networks (NNs) or neural models. In the human brain, neurons are electrically excitable cells and the fundamental component of the nervous system. They store information and communicate with other cells through thousands of synapses. The human brain has approximately 86 billion neurons, and their connections form a vast and complex network [91].

Fig. 2.12 (a) depicts the conceptual structure of a typical neuron. A neuron consists of a cell body (soma), dendrites, and a single axon. The cell membranes of soma contain various receptors and ion channels. These receptors can bind to the corresponding chemical neurotransmitters, which cause changes in ion permeability and a potential difference between the inner and outer membranes. The soma generates physiological activity, including excitation and inhibition. The dendrites receive signals and transmit them to the cell body for signal processing. The processed signal is then sent out by axon.

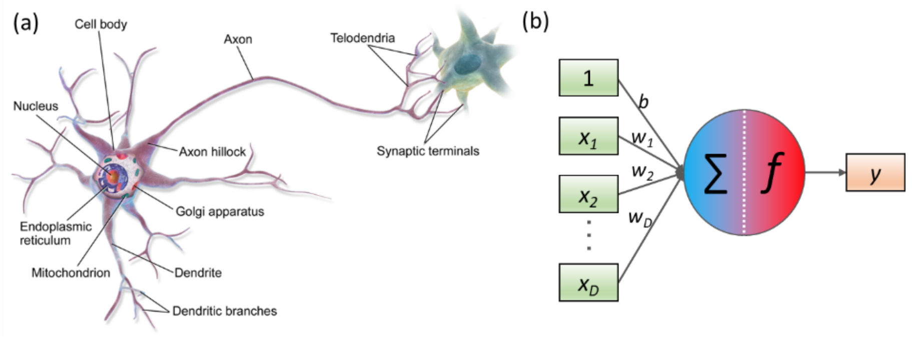


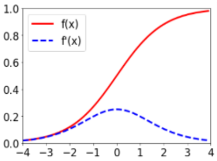
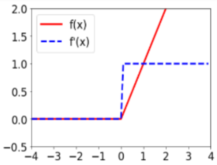
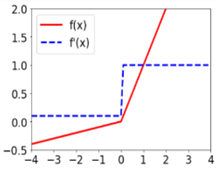
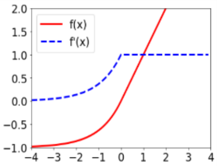
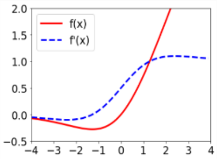
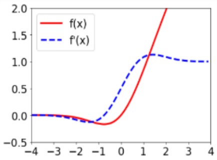
Fig. 2.12 (a) Typical neural structure. (b) Artificial neuron. It accepts *D* inputs and calculates the weighted average. The weighted average is also noted as net input, which is then processed by an activation function *f*. Fig. 2.12 (a) was adopted from Wikipedia, https://en.wikipedia.org/wiki/Neuron.

The artificial neuron shown in Fig 2.12 (b) mimics the structure and function of a neuron. Suppose the input vector ***X*** has *D* features, $\boldsymbol{X} = [x_1; x_2; \dots; x_D]$. The neuron calculates the weighted sum of ***X*** and then processes the result by an activate function *f*:

$$y = f\left(\sum_{d=1}^{D} w_d\, x_d + b\right) = f(w^T x + b), \tag{2.32}$$

where $w = [w_1; w_2; \dots; w_D] \in \mathbb{R}^D$ is a *D*-dimension weight vector, $b \in \mathbb{R}^D$ is the bias value. The activation function is a nonlinear function that can expand the network's representation capability and learning ability. It should be simple enough to improve the computational efficiency, and its value and derivative's value should be in suitable ranges to improve training stability.

Table 2.1 Common nonlinear activation functions

Name	Function	Derivative	Graph
Sigmoid	$f(x) = \frac{1}{1+e^{-x}}$	$f'(x) = \frac{e^x}{(1+e^x)^2}$	
ReLU	$f(x) = \begin{cases} x, & x \geq 0 \\ 0, & x < 0 \end{cases}$	$f'(x) = \begin{cases} 1, & x \geq 0 \\ 0, & x < 0 \end{cases}$	
Leaky-ReLU	$f(x) = \begin{cases} x, & x \geq 0 \\ ax, & x < 0 \end{cases}$	$f'(x) = \begin{cases} 1, & x \geq 0 \\ a, & x < 0 \end{cases}$	
ELU	$f(x) = \begin{cases} x, & x \geq 0 \\ a(e^x - 1), & x < 0 \end{cases}$	$f'(x) = \begin{cases} 1, & x \geq 0 \\ ae^x, & x < 0 \end{cases}$	
Swish	$f(x) = \frac{x}{1+e^{-\beta x}}$	$f'(x) = \frac{\beta x}{1+e^{-\beta x}} + \frac{1}{1+e^{-\beta x}}(1- \frac{\beta}{1+e^{-\beta x}})$	
GELU	$f(x) = x\phi(x)$	$f'(x) = \phi(x) + xP(X = x)$	

Note: $\phi(x)$ is the Cumulative Distribution Function of a standard normal:

$$\phi(x) = P(X \leq x) = \frac{1}{2}\left(1 + erf\left(x/\sqrt{2}\right)\right), X \sim N(0,1)$$

Table 2.1 presents the definitions, derivatives, and graphs of commonly used nonlinear activation functions in ANNs. The sigmoid, a mathematical function with an “S”-shaped curve, was popular in early ANN literature due to its desirable mathematical properties. The sigmoid function is continuously conductive. Its output range is within (0, 1), making it interpretable as probabilities, which is useful in some computations. Tanh and Hard-Tanh are similar activation functions. However, the sigmoid function saturates at both ends, resulting in zero gradients and a gradient vanishing problem, making it unsuitable for deep ANN training. Thus, non-saturating activation functions like piecewise linear activation functions are preferred. One of the most popular piecewise linear activation functions is Rectified Linear Unit (ReLU) [92]. ReLU has high computational efficiency because it only requires addition, multiplication, and comparison operations. It also has reasonable biological plausibility, such as unilateral inhibition, wide excitation boundary, and sparsity. ReLU can alleviate the gradient vanishing problem. However, the output of ReLU is non-zero-centric because ReLU only outputs non-negative values. It will introduce an offset bias and affect the efficiency of gradient descent. In addition, if the weights in some neurons are initialized to be large negative values, then the neurons have a slim chance of being activated. This is the "dying ReLU" problem. In practice, some variants of ReLU are proposed to avoid these problems.

The Leaky-ReLU attempts to fix the "dying ReLU" problem by introducing a slight negative slope when the input value is negative [93]. The negative slope can also adjust through network training. In this case, it is called Parametric ReLU (PReLU). To address the offset bias problem, Exponential Linear Unit (ELU) is developed [94]. ELU has approximately zero centralized output. The Swish function is a self-gated activation function, which is a non-linear interpolation function between the linear function and ReLU [95]. The parameter β controls the shape of the Swish function. The Gaussian Error Linear Unit (GELU) is also a gate-controlled activation function, like Swish [96].

Although a single neuron has a simple function, many neurons work together to form ANNs, realizing very complex functions. ANNs are large-scale parallel distributed processing systems in which the neurons connect via a specific communication topology. The ANN has three features:

(a) The representation of information is non-local and distributed throughout the whole network.

(b) Memory and knowledge are stored in the connections between neurons.

(c) Learning is done by changing the weights of connections between neurons.

A complete ANN algorithm contains three parts: network architectures (models), backpropagation and loss function, and network optimization and regularization. The following sections will explain them in more detail.

2.6 Neural network architectures

The contents in this section are mainly summarized from [97] and [98].

2.6.1 Feedforward Neural Networks

The Feedforward Neural Network (FNN) is the earliest developed simple network architecture. It is also named Multi-Layer Perceptron (MLP). Fig. 2.13 shows an example of FNN. The first layer (input layer) receives the input signal. Then the signal passes through several hidden layers and is processed by neurons inside each layer. The latest layer is the output layer, sending out the final signal. The whole network has no feedback and loop, and the signal propagates unidirectionally from the input layer to the output layer. Therefore, FNN can be described by a directed acyclic graph (DAG).

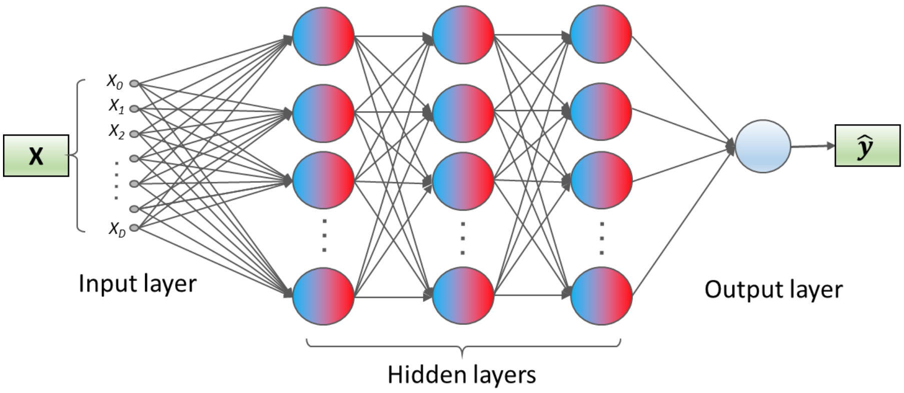


Fig. 2.13. Feedforward Neural Network (FNN) architecture.

Let $\boldsymbol{X}$ be the input signal, $\hat{\boldsymbol{y}}$ the output signal, L the layer number, M_l the neuron number in the l^{th} layer, $\mathbf{W}^{(l)} \in \mathbb{R}^{M_l \times M_{l-1}}$the weight matrix of the l^{th} layer, $\boldsymbol{b}^{(l)} \in \mathbb{R}^{M_l}$ the bias of l^{th} layer, and f_l the activation function of l^{th} layer. The FNN can be expressed as a nested function:

$$\hat{\boldsymbol{y}} = \boldsymbol{f}_L(\boldsymbol{W}^{(L)}(\ldots f_2(\boldsymbol{W}^{(2)} f_1(\boldsymbol{W}^{(1)}\boldsymbol{X} + \boldsymbol{b}^{(1)}) + \boldsymbol{b}^{(2)}) + \boldsymbol{b}^{(L-1)}) + \boldsymbol{b}^{(L)}). \qquad (2.33)$$

It is easy to see that each layer implements an affine transformation and a nonlinear transformation on the signal from the previous layer. The whole FNN can also be described as:

$$\hat{\boldsymbol{y}} = \boldsymbol{f}(\boldsymbol{X}; \boldsymbol{W}, \boldsymbol{b}), \qquad (2.34)$$

where $\boldsymbol{W}$ and $\boldsymbol{b}$ denote the weight matrices and biases of all layers.

According to the Universal Approximation Theorem, FNN has a compelling ability to approximate any bounded and closed-set function with arbitrary precision [99]. However, in practice, such capability is prone to overfitting. In addition, we often do not know the underlying mapping functions and how to find the optimized FNN architectures and parameters. Instead, the neural network should be trained to learn from the training dataset by minimizing the pre-defined loss functions using some optimization algorithms and regularizations.

2.6.2 Convolutional Neural Networks

In practical tasks, many data are highly structured. For example, voice signals have one-dimensional (1D) temporal structures, natural images have 2D spatial structures, and time-resolved images have 3D spatial-temporal structures. FNN has some notable disadvantages when processing such structured data. First, FNN tends toward oversized parameters. Take a 2D image $\boldsymbol{X} \in \mathbb{R}^{H \times W \times C}$ as an example, where H, W, and C are the width, height, and color channel, respectively. If an FNN has L layers and each layer has N neurons, there will be $H \times W \times C \times L \times N$ connections in total. The number of parameters is enormous and will dramatically increase as network size increases, leading to difficulty in network training and overfitting problems. Second, in natural images, many objects have local invariance features. Their semantic information is not affected by scale, shift, and rotation operations. FNN cannot capture the local invariance features. The convolutional neural network (CNN) is designed to address these problems. CNN is inspired by the receptive field in the brain’s visual cortex, which is a specific FNN with a local connection, weight sharing, and information pooling features. CNN has been the dominant network architecture in imaging tasks, achieving outstanding successes in imaging tasks such as image classification, segmentation, object detection, human face recognition, and so on. The accuracy significantly outperforms other network architectures. It has also found various applications in natural language processing, reinforcement learning, and recommendation systems.

In CNN, the matrix multiplication is replaced by a convolution operation. The 1D convolution between two discrete vectors $\boldsymbol{x} \in \mathbb{R}^L$, and $\boldsymbol{w} \in \mathbb{R}^N$ is described as:

$$y_t = \sum_{n=1}^{N} w_k \, x_{t-k+1}, \tag{2.35}$$

or:

$$\boldsymbol{y} = \boldsymbol{w} * \boldsymbol{x}. \tag{2.36}$$

The asterisk * denotes the convolution operator. The vector $\boldsymbol{w}$ is often named filter or convolution kernel. Likewise, the 2D convolution between an image $\boldsymbol{x} \in \mathbb{R}^{M \times N}$ and a filter $\boldsymbol{w} \in \mathbb{R}^{U \times V}$ is defined as:

$$y_{ij} = \sum_{u=1}^{U} \sum_{v=1}^{V} w_{uv} \, x_{i-u+1,j-v+1}, \tag{2.37}$$

or:

$$\boldsymbol{Y} = \boldsymbol{W} * \boldsymbol{X}. \tag{2.38}$$

It should be noted that in the deep learning literature, the convolution operator is replaced by the cross-correlation operator. The cross-correlation is closely related to convolution, in which the filter $\boldsymbol{w}$ does not need to flip for the multiplication. The 2D cross-correlation is defined as:

$$y_{ij} = \sum_{u=1}^{U} \sum_{v=1}^{V} w_{uv} \, x_{i+u-1,j+v-1}. \tag{2.39}$$

It can be denoted as:

$$\boldsymbol{Y} = \boldsymbol{W} \otimes \boldsymbol{X}. \tag{2.40}$$

Because in CNN, the parameters in $\boldsymbol{w}$ are obtained through learning, the convolution is equivalent to cross-correlation. It is unnecessary to add additional operations to flip the filter. Thus, all convolution is cross-correlation by default in CNNs.

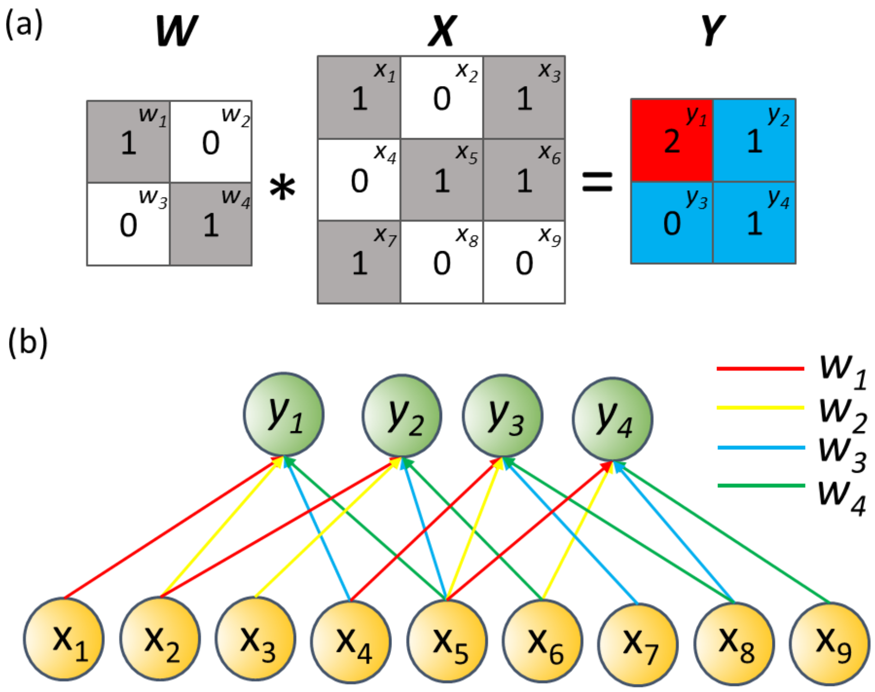


Fig. 2.14. (a) schematic view of a 2D convolution between 2×2 and 3×3 matrices. (b) The equivalent FNN of the 2D convolution.

Fig. 2.14 shows a simple example of 2D convolution, $\boldsymbol{Y} = \boldsymbol{W} \otimes \boldsymbol{X}$, where $\boldsymbol{W}$ and $\boldsymbol{X}$ are 2×2 and 3×3 matrices, respectively. Fig. 2.14 (a) shows convolution as an effective method for feature extraction. The filter works as a template sliding through all input image patches. After the convolution, a feature map that matches the filter is obtained. We usually use many filters to extract different feature maps. Fig. 2.14 (b) shows the equivalent FNN of the convolution. The neurons are locally connected and share weight parameters in different connections. In addition, the network becomes sparse compared with fully connected FNNs. From this viewpoint, CNN is a highly regularized FNN.

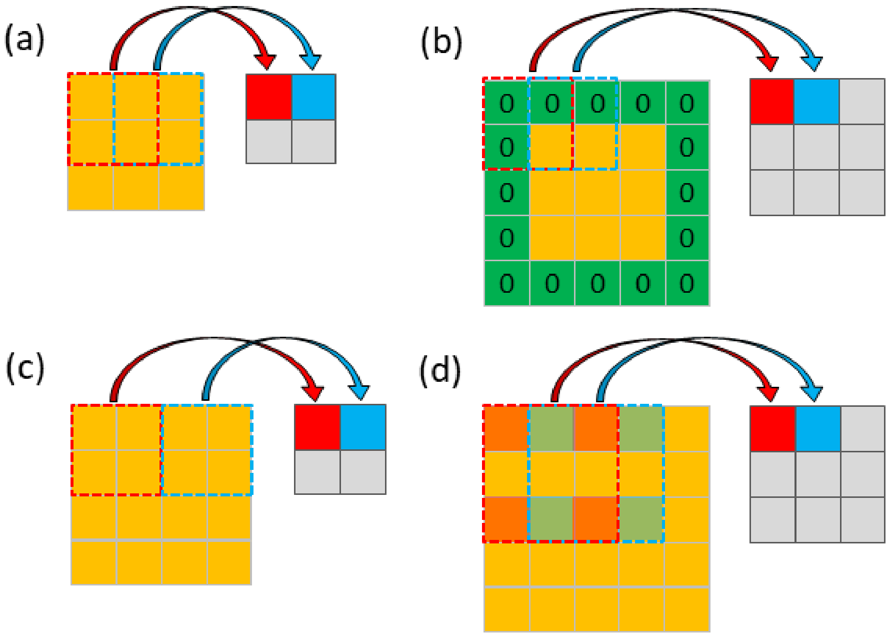


Fig. 2.15. Schematic view of the sliding window in a 2D convolutional operation with (a) stride = 1, (b) zero padding on the boundaries, (c) stride = 2, and (d) dilation rate = 2.

The convolution layers have various configurations by setting the stride, padding, and dilation. Fig. 2.15 (a) shows an example of 2D convolution between a 3×3 image and a 2×2 kernel with stride = 1. The output matrix is 2×2. This type of convolution is called valid convolution. Fig. 2.15 (b) shows the same convolution with "0" padding applied on the boundaries. The output has the same size as the matrix. This type of convolution is called the same convolution. The value in the padding pixels and padding pixel number can be set arbitrarily. Fig. 2.15 (c) shows the convolution with stride = 2. The kernel slides the image by skipping every 2 pixels. Fig. 2.15 (d) shows the dilated convolution with dilation rate = 2. Dilated convolution enlarges the receptive field while keeping a few parameters in the kernel. For dilation rate = r, it takes every r^{th} value in the input image when performing convolution. Other pixels in the kernel are set to zero. Therefore, dilated convolution is also called convolution with holes. In a general 2D convolution, if the padding number is (p_1,p_2), stride is (s_1,s_2), kernel size is (k_1,k_2), and dilation rate is (d_1,d_2), for a H×W input image, the size of the output image is:

$$H' = \left\lfloor \frac{H + 2p_1 - d_1(k_1 - 1) - 1}{s_1} + 1 \right\rfloor, \tag{2.41}$$

$$W' = \left\lfloor \frac{W + 2p_2 - d_2(k_2 - 1) - 1}{s_2} + 1 \right\rfloor, \tag{2.42}$$

where $\lfloor \cdot \rfloor$ denotes the rounding down operator.

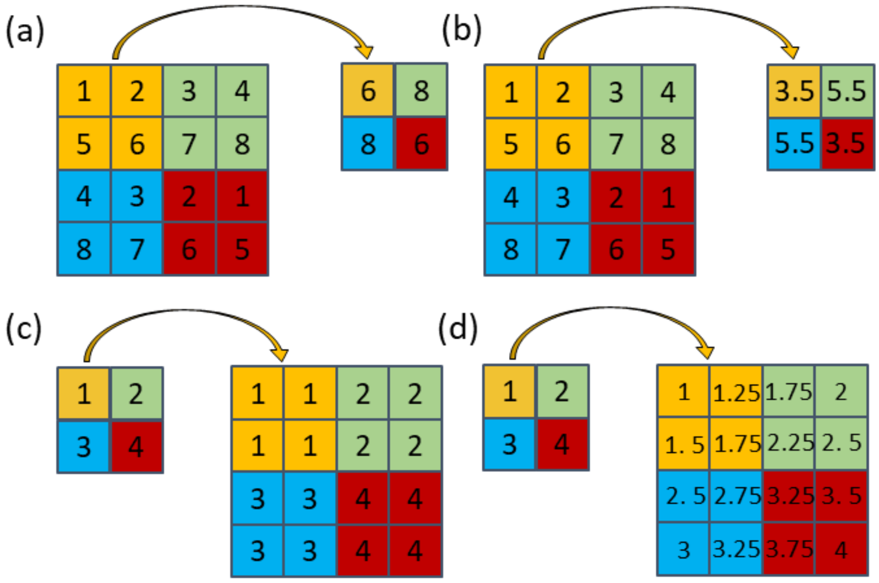


Fig. 2.16. Pooling layers and up-sampling layers: (a) max pooling, (b) average pooling, (c) up-sampling layer by repeating input values, and (d) up-sampling layer by interpolating the selected areas.

The convolution layers often work together with the pooling layers, as illustrated in Fig. 2.16 (a) and (b). The pooling layers, also called the subsampling layers, are used for feature selection. There are two commonly used functions in pooling layers: max pooling and average pooling. Fig. 2.16 (a) shows the max pooling, which uses the maximum value of the selected area as the output. The average pooling is an alternative operation that calculates the average values of the selected area as the output. In contrast to pooling layers, the up-sampling layers increase the features by repeating the input values (Fig. 2.16 (c)) or interpolating the selected areas (Fig. 2.16 (d)).

The batch normalization (BN) layers are effective techniques for designing deep networks. As mentioned above, deep networks are difficult to train because of the gradient vanishing and exploding problems. BN layers provide an effective solution to alleviate these problems and make the training easier by standardizing the distribution of layer outputs with zero mean and unit variance in a minibatch. Assume the output vector after the activation function is z_n and n is the batch size. BN layer calculates the output $\hat{z}_n$ as follows:

$$\hat{z}_n = \gamma \odot \frac{z_n - \boldsymbol{\mu}_{\mathcal{B}}}{\sqrt{\boldsymbol{\sigma}_{\mathcal{B}}^2 + \epsilon}} + \beta, \tag{2.43}$$

$$\boldsymbol{\mu}_{\mathcal{B}} = \frac{1}{|\mathcal{B}|} \sum_{z \in \mathcal{B}} z, \tag{2.44}$$

$$\boldsymbol{\sigma}_{\mathcal{B}}^2 = \frac{1}{|\mathcal{B}|} \sum_{z \in \mathcal{B}} (z - \boldsymbol{\mu}_{\mathcal{B}})^2 . \tag{2.45}$$

In the above functions, $\mathcal{B}$ is the minibatch with n samples; $\boldsymbol{\mu}_{\mathcal{B}}$ and $\boldsymbol{\sigma}_{\mathcal{B}}^2$ are the mean and variance of the outputs for the minibatch; β and γ are learnable parameters in the BN layer, and $\epsilon > 0$ is a small constant to keep numerical stability.

There are various network architectures for different tasks by assembling different kinds of layers. Fig. 2.17 shows the schematic views of four typical backbone network architectures. The VGG16 shown in Fig. 2.17 (a) is a popular network architecture for image classification, achieving 92.7% accuracy in classifying 1000 images of 1000 different categories [100]. VGG16 is considered a milestone in computer vision. There are 21 layers in VGG16, including thirteen convolution layers, five max-pooling layers, and three dense layers. The "16" means 16 trainable layers (convolution and dense layers). VGG16 takes 224×224 images with 3 RGB channels as inputs. The basic building block contains a stack of convolution layers with a 3×3 kernel size, one stride, and one zero padding, followed by a max-pooling layer. Three fully connected layers follow the convolution layers. The output layer has a Softmax activation and outputs the possibilities of 1000 classes. The outstanding performance of VGG16 is attributed to the use of small convolution kernels, which can be seen as a regularization of large kernel size and significantly reduces model parameters. Fig. 2.17 (b) shows the architecture of the Auto-Encoder [101]. An Auto-Encoder contains an encoder and a decoder, which have a symmetric structure and form a bottleneck shape. The encoder learns the representation of the input and compresses and maps the input to latent variables, whereas the decoder reconstructs the original input. Auto-Encoders are popular in both supervised and unsupervised representation learning tasks. The U-net shown in Fig. 2.17 (c) can be seen as a variant of the auto-encoder [102]. There are skip connections between different levels of the down-sampling and up-sampling layers, improving the effective usage of information. U-net has achieved superior performance in biomedical image segmentation. It is also a popular architecture for various end-to-end tasks. Fig. 2.17 (d) shows a general model of Generative adversarial networks (GANs) [103]. GANs belong to the family of generative models, which can generate new data that resemble the training data. A GAN contains a generative model and a discriminative model. The generative model creates new data with similar distributions to the training data. In contrast, the discriminative model tries to determine whether the input is generated by the generative model or from the training data. They train and compete with each other until they reach Nash equilibrium. GANs have found various applications, including image generation, 3D model reconstruction, and data argumentation, to name a few [104]. One challenge for GANs is the difficulty in network training. Model collapse and non-convergence problems often happen during model training.

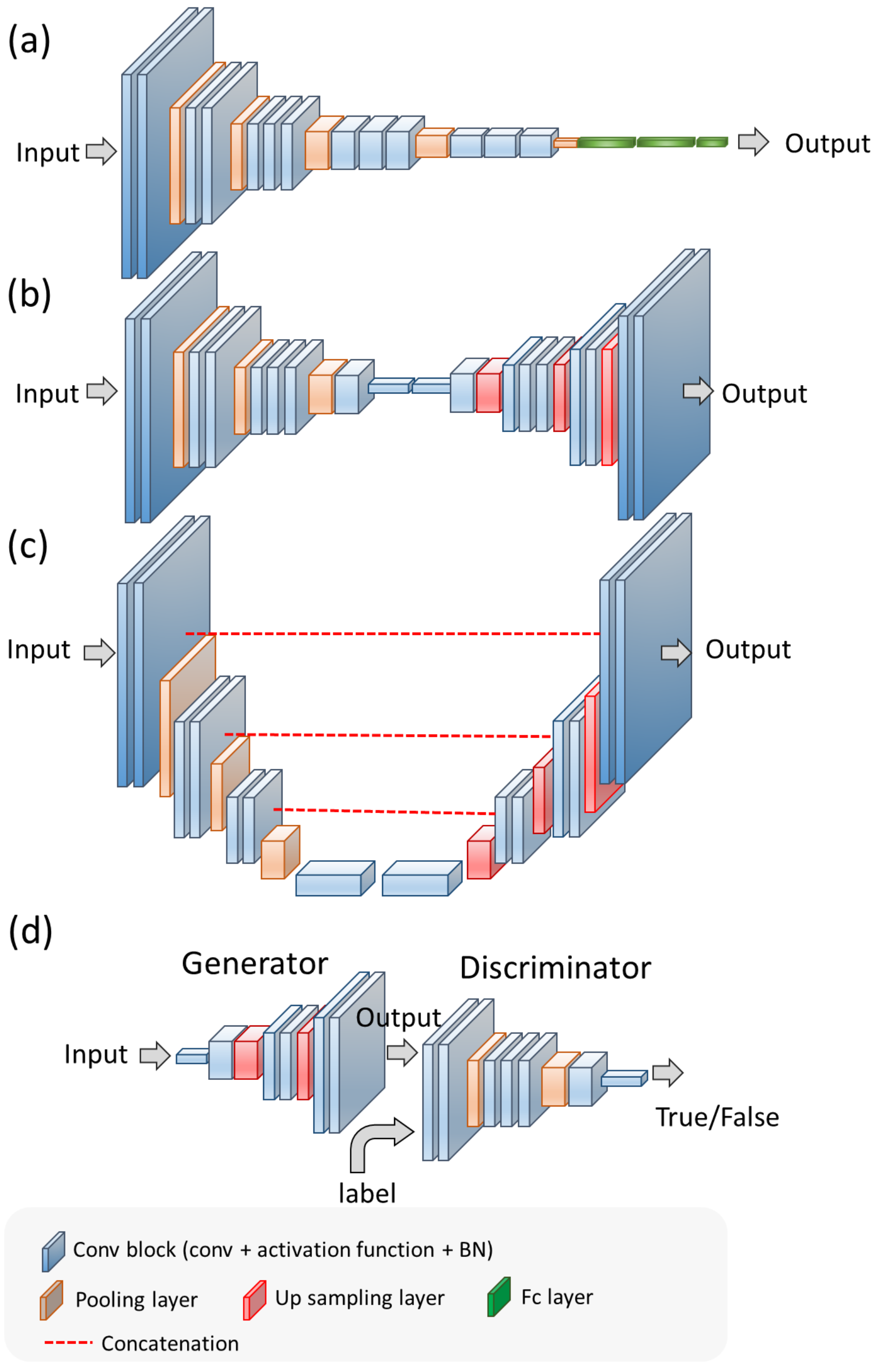


Fig. 2.17. Common DL architectures. (a) VGG16, (b) Auto-Encoder, (c) U-net, and (d) GAN.

2.6.3 *Recurrent Neural Network (RNN)*

Both FNN and CNN are complex nested functions. The output depends on the input, and the information unidirectionally transmits from the head to the toe. Although this “feed-forward” architecture makes the network easy to train, it limits the network’s capability at the same time.

In many tasks for sequences such as machine translation, natural language processing, and text classification, the output depends not only on the input but also on the previous state/output. In addition, FNN and CNN have fixed-length input and output, being difficult to process length-variant time-series data such as video, text, and audio.

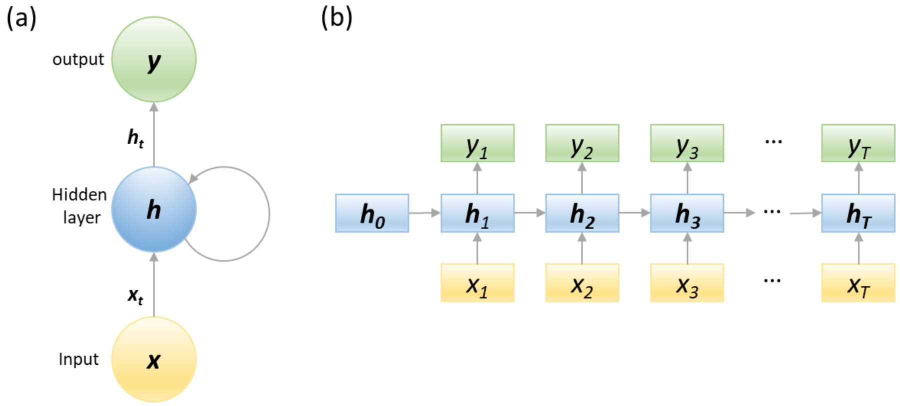


Fig. 2.18. RNN and its unrolled form in time sequence.

Recurrent Neural Networks (RNNs) are designed for sequence processing and can address the above problems. An RNN has short-term memory and loop structures so that the neuron in RNN can receive information from both other neurons and the neuron itself. Fig. 2.18 (a) shows a simple RNN with one hidden layer. The hidden layer has a feedback connection. Assume the input is $\boldsymbol{x}_t \in \mathbb{R}^M$ and the hidden state is $\boldsymbol{h}_t \in \mathbb{R}^D$ at time step *t*. The update of the hidden layer at time step *t* is determined by the input of the current time step together with the previous hidden state:

$$\boldsymbol{h}_t = \boldsymbol{f}(\boldsymbol{W}_x\boldsymbol{x}_t + \boldsymbol{W}_h\boldsymbol{h}_{t-1} + \boldsymbol{b}). \tag{2.46}$$

In Eq. (2.46), $\boldsymbol{W}_x \in \mathbb{R}^{D\times M}$ and $\boldsymbol{W}_h \in \mathbb{R}^{D\times D}$ are the weight matrixes for $\boldsymbol{x}_t$ and $\boldsymbol{h}_{t-1}$, respectively. $\boldsymbol{b} \in \mathbb{R}^D$ is the bias matrix. The activation function of the hidden layer is $\boldsymbol{f}(\cdot)$. Fig. 2.18(b) illustrates the computational logic of the unrolled RNN at sequenced time steps.

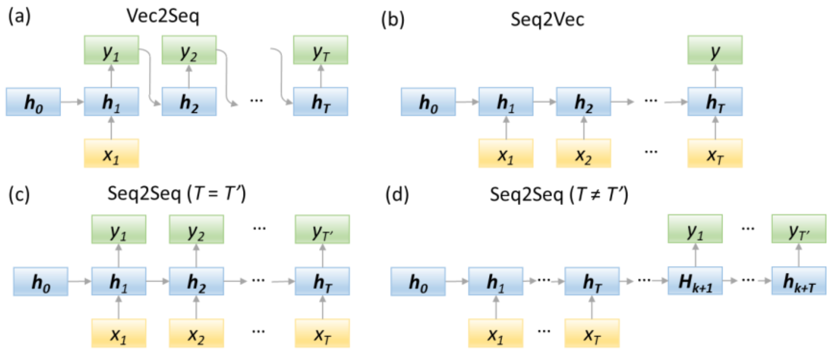


Fig. 2.19 variants of RNNs

Different variants of RNNs are summarized in Fig. 2.19. The Vec2Seq model in Fig. 2.19 (a) is used for sequence generation. The input is a vector, and the output is an arbitrary-length sequence of vectors. Vec2Seq models are useful for language modeling, music generation, and image captioning. On the contrary, the Seq2Vec model in Fig. 2.19 (b) receives a sequence of vectors as input and outputs a vector. Seq2Vec models are widely used in sequence classifications. The Seq2Seq model takes sequences as both input and output. There are two cases for Seq2Seq models shown in Fig. 2.19 (c) and (d). The first is aligned Seq2Seq models, in which the input and output sequences have the same length. These models are mainly used for sequence labeling, such as part-of-speech tagging and name entity recognition. The second is unaligned Seq2Seq models. The input and output sequences have different lengths. Unaligned Seq2Seq models are also called encoder-decoder models. They have important applications in machine translation.

Theoretically, simple RNNs can learn the long-term dependencies among time states, given that RNNs have sufficient layers. However, only short-term dependencies are learned due to gradient exploding and vanishing problems. To address the long-term dependencies problem, Gated RNNs use the gating mechanism to control the flow of information. There are two typical gated architectures to maintain a separate cell state: Long Short-Term Memory Network (LSTM) and Gated Recurrent Unit (GRU).

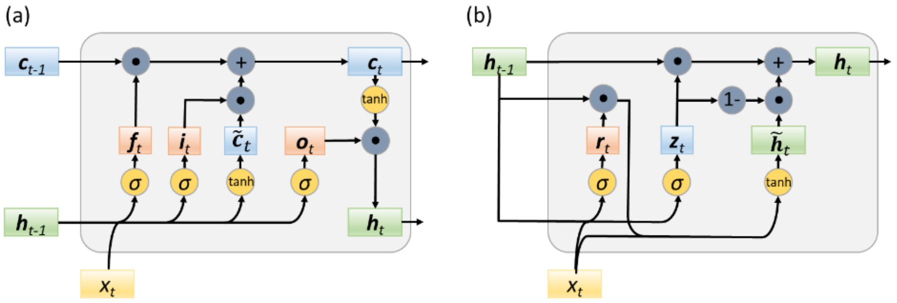


Fig. 2.20. LSTM and GRU architectures.

In simple RNNs, the hidden states can be viewed as short-term memory as they are re-written every time. In contrast, in LSTMs, the memory cell can capture important information at some time and keep it for a relatively long period. It is the reason why such architecture is called Long Short-Term Memory Network. As shown in Fig. 2.20 (a), the basic idea of LSTMs is to control an internal state in a cell using three gates: the input gate $\boldsymbol{i}_{\mathrm{t}} \in [0,1]^D$ controls the input information, the forget gate $\boldsymbol{f}_{\mathrm{t}} \in [0,1]^D$ determines what information should be reset, and the output gate $\boldsymbol{o}_{\mathrm{t}} \in [0,1]^D$ control the output information. All three gates are soft gates with values ranging from 0 to 1. They update as follows:

$$\boldsymbol{i}_t = \sigma(\boldsymbol{W}_i \boldsymbol{x}_t + \boldsymbol{U}_i \boldsymbol{h}_{t-1} + \boldsymbol{b}_i), \tag{2.47}$$

$$\boldsymbol{f}_t = \sigma\left(\boldsymbol{W}_f \boldsymbol{x}_t + \boldsymbol{U}_f \boldsymbol{h}_{t-1} + \boldsymbol{b}_f\right), \tag{2.48}$$

$$\boldsymbol{o}_t = \sigma(\boldsymbol{W}_o \boldsymbol{x}_t + \boldsymbol{U}_o \boldsymbol{h}_{t-1} + \boldsymbol{b}_o), \tag{2.49}$$

where is the logistic function with an output range [0, 1]. Then a candidate state is computed as:

$$\widetilde{\boldsymbol{C}}_t = tanh(\boldsymbol{W}_c \boldsymbol{x}_t + \boldsymbol{U}_c \boldsymbol{h}_{t-1} + \boldsymbol{b}_c). \tag{2.50}$$

The actual update of the state is:

$$\boldsymbol{C}_t = \boldsymbol{f}_t \odot \boldsymbol{C}_{t-1} + \boldsymbol{i}_t \odot \widetilde{\boldsymbol{C}}_t. \tag{2.51}$$

The final output is:

$$\boldsymbol{h}_t = \boldsymbol{o}_t \odot \boldsymbol{tanh}(\boldsymbol{C}_t). \tag{2.52}$$

The GRU is a simpler architecture than LSTMs. Fig. 2.20 (b) shows the architecture of GRU. The update of GRU is:

$$\boldsymbol{h}_t = \boldsymbol{z}_t \odot \boldsymbol{h}_{t-1} + (1 - \boldsymbol{z}_t) \odot \widetilde{\boldsymbol{h}}_t. \tag{2.53}$$

In the above equation, $\boldsymbol{z}_{\mathrm{t}} \in [0,1]^D$ is the update gate, which is:

$$\boldsymbol{z}_t = \sigma(\boldsymbol{W}_z\boldsymbol{x}_t + \boldsymbol{U}_z\boldsymbol{h}_{t-1} + \boldsymbol{b}_z). \tag{2.54}$$

$\widetilde{\boldsymbol{h}}_t$ is the candidate state at time *t*, which is:

$$\widetilde{\boldsymbol{h}}_t = tanh(\boldsymbol{W}_h\boldsymbol{x}_t + \boldsymbol{U}_h(\boldsymbol{r}_t \odot \boldsymbol{h}_{t-1}) + \boldsymbol{b}_h), \tag{2.55}$$

where $\boldsymbol{z}_{\mathrm{t}} \in [0,1]^D$is the reset gate, which is:

$$\boldsymbol{r}_t = \sigma(\boldsymbol{W}_r\boldsymbol{x}_t + \boldsymbol{U}_r\boldsymbol{h}_{t-1} + \boldsymbol{b}_r). \tag{2.56}$$

When $z_t = 0$ and $\boldsymbol{r}_t = 1$, the GRU degrades to a simple RNN; When $z_t = 0$ and $\boldsymbol{r}_t = 0$, the current state $\boldsymbol{h}_{\mathrm{t}}$ only relates to the current input $\boldsymbol{x}_t$; When $z_t = 0$, the current state $\boldsymbol{h}_{\mathrm{t}}$ equals the previous state $\boldsymbol{h}_{\mathrm{t-1}}$ and is independent to the input $\boldsymbol{x}_t$.

2.6.4 Attention Mechanism and Transformer

The attention mechanism mimicking the cognitive mechanism in human brains can selectively process useful information and omit irrelevant information, thus significantly improving the computational efficiency and neural networks' capability [105]. In general, the input information can be represented by Key-Value pairs. Then we introduce Query vectors to calculate the similarity between keys and queries and dynamically decide which values should be used. It is the basic idea of the attention mechanism. Considering *N* input Key-Value pairs $(\boldsymbol{K}, \boldsymbol{V}) = [(\boldsymbol{k_1}, \boldsymbol{v_1}), \cdots, (\boldsymbol{k_N}, \boldsymbol{v_N})]$, and the query vector $\boldsymbol{q}$, the attention function is:

$$att\big((\boldsymbol{K}, \boldsymbol{V}), \boldsymbol{q}\big) = \sum_{n=1}^{N} \alpha_n \boldsymbol{v_n} = \sum_{n=1}^{N} \frac{exp\big(s(\boldsymbol{k_n}, \boldsymbol{q})\big)}{\sum_j exp\Big(s\big(\boldsymbol{k_j}, \boldsymbol{q}\big)\Big)} \boldsymbol{v_n}\,. \tag{2.57}$$

In the above function, $s(\boldsymbol{k}, \boldsymbol{q})$ is the scoring function. There are two steps in the calculation of attention functions. The first step is to calculate the attention distribution α_n using a scoring function $s(\cdot)$. The second step is to calculate the weighted average of the input values. Two typical scoring functions are used for calculating α_n, the addictive function and the scaled dot-product function. The addictive function is:

$$s(\boldsymbol{k}, \boldsymbol{q}) = \boldsymbol{W}^T tanh\left(\boldsymbol{W}_x \boldsymbol{x} + \boldsymbol{W}_q \boldsymbol{q}\right), \tag{2.58}$$

where $\boldsymbol{W}$, $\boldsymbol{W}_x$, and $\boldsymbol{W}_q$ are learnable parameters. The scaled dot-product function is:

$$s(\boldsymbol{k}, \boldsymbol{q}) = \frac{\boldsymbol{k}^T \boldsymbol{q}}{\sqrt{d}}, \tag{2.59}$$

where *d* is the dimension of the input vector. When *d* is larger, the dot-product has a large variance, leading to a small gradient of the softmax function. The denominator $\sqrt{d}$ can effetely alleviate this problem. Both addictive and scaled dot-product functions have similar computational complexity. The scaled dot-product function is more prevailing because it takes advantage of matrix multiplication well and has better computational efficiency. The attention can be extended to multi-head attention, in which we use multiple query vectors $\boldsymbol{Q} = [\boldsymbol{q_1}, \cdots, \boldsymbol{q_M}]$ to choose different information from the input:

$$att\left((\boldsymbol{K}, \boldsymbol{V}), \boldsymbol{Q}\right) = att\left((\boldsymbol{K}, \boldsymbol{V}), \boldsymbol{q}_1\right) \oplus, \cdots, \oplus\, att\left((\boldsymbol{K}, \boldsymbol{V}), \boldsymbol{q}_M\right), \tag{2.60}$$

where $\oplus$ denotes vector concatenation. The attention mechanism techniques have improved the performance of various neural networks and become an omnipresent component in state-of-the-art models.

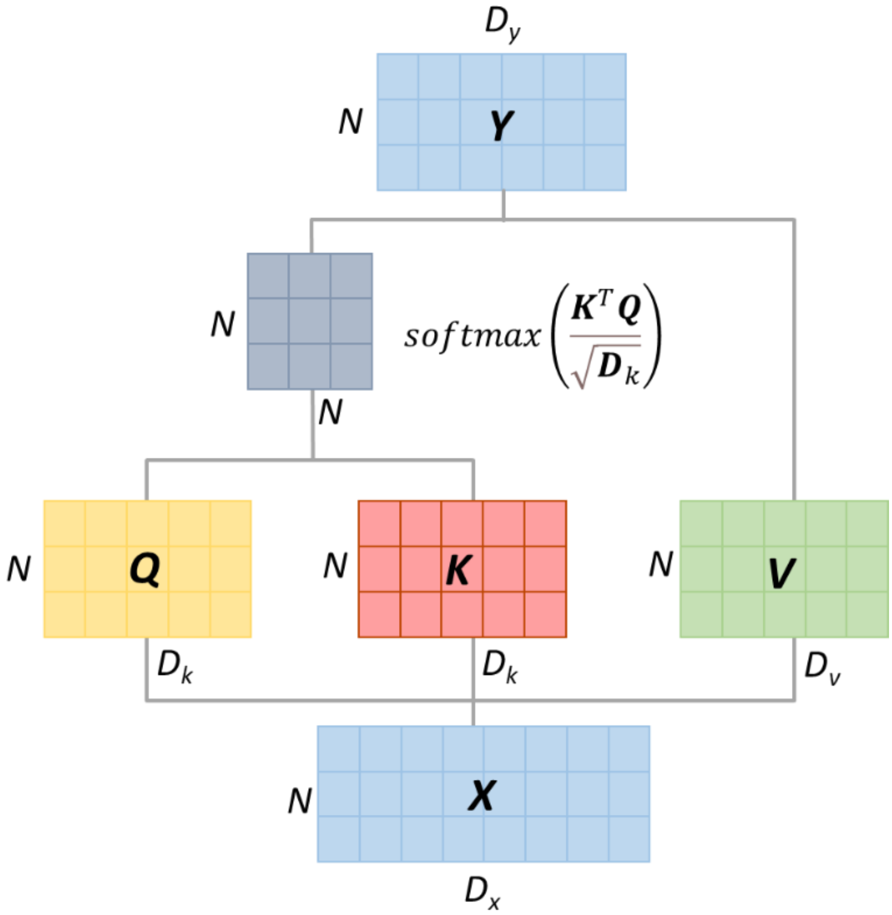


Fig.2.21. Calculation process in the self-attention model.

The self-attention models automatically utilize the attention mechanism to generate Query-Key-Value pairs from input data. Fig. 2.21 shows the calculation process of the self-attention model. Given a sequence of input $\boldsymbol{X} = [\boldsymbol{x}_1, \cdots, \boldsymbol{x}_N] \in \mathbb{R}^{D_x \times N}$, the self-attention model generates the output $\boldsymbol{Y} = [\boldsymbol{y}_1, \cdots, \boldsymbol{y}_N] \in \mathbb{R}^{D_y \times N}$. Firstly, the input linearly maps to $\boldsymbol{Q} = [\boldsymbol{q}_1, \cdots, \boldsymbol{q}_N]$, $\boldsymbol{K} = [\boldsymbol{k}_1, \cdots, \boldsymbol{k}_N]$, and $\boldsymbol{V} = [\boldsymbol{v}_1, \cdots, \boldsymbol{v}_N]$ by multiplying weight matrixes:

$$\begin{aligned} \boldsymbol{Q} &= \boldsymbol{W}_q \boldsymbol{X} \in \mathbb{R}^{D_k \times N}, \\ \boldsymbol{k} &= \boldsymbol{W}_k \boldsymbol{X} \in \mathbb{R}^{D_k \times N}, \\ \boldsymbol{V} &= \boldsymbol{W}_v \boldsymbol{X} \in \mathbb{R}^{D_v \times N}, \end{aligned} \tag{2.61}$$

where $\boldsymbol{W}_q \in \mathbb{R}^{D_k \times D_x}$, $\boldsymbol{W}_k \in \mathbb{R}^{D_k \times D_x}$, and $\boldsymbol{W}_v \in \mathbb{R}^{D_v \times D_x}$ are weight matrixes for $\boldsymbol{Q}$, $\boldsymbol{K}$, and V (Note $D_v = D_y$). Then for a single $\boldsymbol{q}_n$, the output $\boldsymbol{y}_n$ is:

$$\boldsymbol{y}_n = att\big((\boldsymbol{K},\boldsymbol{V}),\boldsymbol{q}_n\big) = \sum_{j=1}^{N} softmax\left(s\big(\boldsymbol{k}_j,\boldsymbol{q}_n\big)\right)\boldsymbol{v}_j\,. \tag{2.62}$$

If the scaled dot-product function is used as the scoring function, the output Y is:

$$\boldsymbol{Y} = \boldsymbol{V}softmax\left(\frac{\boldsymbol{K}^T\boldsymbol{Q}}{\sqrt{D_k}}\right). \tag{2.63}$$

One prominent advantage of the self-attention model is the global field of view for the input vectors. In previous CNNs and RNNs, the input vectors are locally connected. The self-attention model simultaneously considers all input vectors during the calculation, providing an efficient way to solve the long-term dependencies problem. Meanwhile, the computation can be parallelized, and the matrix multiplication is more efficient. The self-attention model can be used as a single layer to replace convolution and recurrent layers or mixed with other layers. The self-attention model can extend to a multi-head attention model using multiple attention matrices to capture different features.

The transformer model uses self-attention mechanisms to construct both the encoder and decoder [105]. The transformer model breaks the barrier between two main areas in DL applications, natural language processing and computer vision. The transformer has outperformed RNNs and achieved state-of-the-art performances in various sequence generation tasks, including machine translation, text generation, and protein sequence generation. It also achieved outstanding success in computer vision tasks such as image recognition, image generation, image Classification, and object detection within a short time [106].

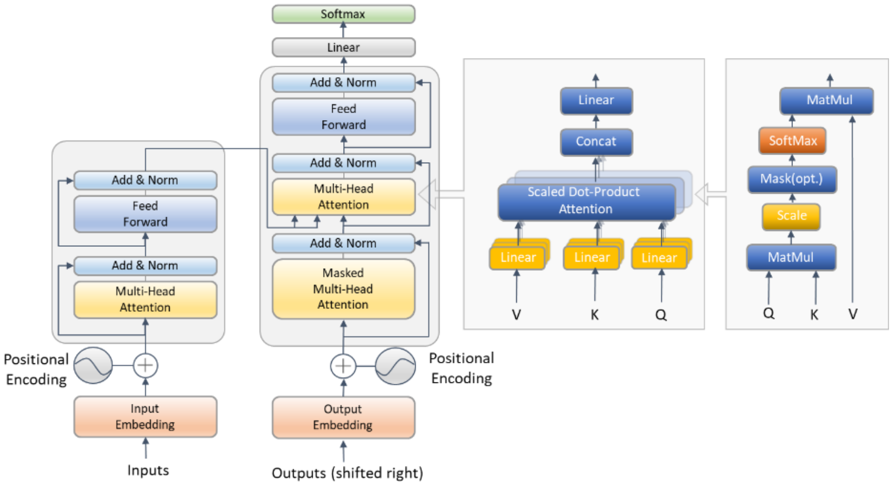


Fig. 2.22 The Transformer model architecture.

Fig. 2.22 shows the architecture of the Transformer. The encoder adopts a series of encoder blocks. Each block uses a multi-headed attention layer, residual connections, and layer normalization. The decoder has similar architecture. It receives both information from the encoder and previously generated outputs. Since the self-attention layers only consider the similarity of input keys and queries and neglect the position information, both input and output should be added with a positional embedding. The positional embedding can be realized by assigning a unique positional vector to each position. Such positional vectors can be manually designed or learned from data. The transformer model can be viewed as a complex version of CNN. In CNN, self-attention can only attend in a fixed receptive field of a convolutional kernel. The transformer extends this ability to learnable receptive fields. In addition, the transformer can also be viewed as a variant of an RNN.

2.7 DL training techniques

2.7.1 Backpropagation and loss function

The loss function should be pre-defined to train the neural network. Then the learning process relies on the backpropagation (BP) algorithm to optimize network parameters. For simplification, Fig. 2.23 shows a feedforward model with a 4-node linear chain to illustrate the BP algorithm.

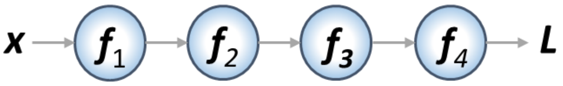


Fig. 2.23. Chain rule for backpropagation algorithm

Assume that the input value is $\boldsymbol{x} \in \mathbb{R}^n$, a composite function $\boldsymbol{f}$ is defined as: $\boldsymbol{f}(\boldsymbol{x}) = (\boldsymbol{f}_4 \circ \boldsymbol{f}_3 \circ \boldsymbol{f}_2 \circ \boldsymbol{f}_1)(\boldsymbol{x})$, where $\boldsymbol{f}_1: \mathbb{R}^n \to \mathbb{R}^{m_1}$, $\boldsymbol{f}_2: \mathbb{R}^{m_1} \to \mathbb{R}^{m_2}$, $\boldsymbol{f}_3: \mathbb{R}^{m_2} \to \mathbb{R}^{m_3}$, $\boldsymbol{f}_4: \mathbb{R}^{m_3} \to \mathbb{R}^{m}$. First, the forward propagation calculates and stores the intermediate variables and output. The intermediate variables are $\boldsymbol{x}_2 = \boldsymbol{f}_1(\boldsymbol{x})$, $\boldsymbol{x}_3 = \boldsymbol{f}_2(\boldsymbol{x}_2)$, $\boldsymbol{x}_4 = \boldsymbol{f}_3(\boldsymbol{x}_3)$, and the output $\boldsymbol{L} = \boldsymbol{f}_4(\boldsymbol{x}_4)$. Second, the backpropagation calculates the gradients using the chain rule from the output to the input:

$$\frac{\partial \boldsymbol{L}}{\partial \boldsymbol{x}} = \frac{\partial \boldsymbol{L}}{\partial \boldsymbol{x}_4}\frac{\partial \boldsymbol{x}_4}{\partial \boldsymbol{x}_3}\frac{\partial \boldsymbol{x}_3}{\partial \boldsymbol{x}_2}\frac{\partial \boldsymbol{x}_2}{\partial \boldsymbol{x}} = \boldsymbol{J}_{f_4}(\boldsymbol{x}_4)\boldsymbol{J}_{f_3}(\boldsymbol{x}_3)\boldsymbol{J}_{f_2}(\boldsymbol{x}_2)\boldsymbol{J}_{f_1}(\boldsymbol{x}_1), \tag{2.64}$$

where $\boldsymbol{J}_{\boldsymbol{f}}(\boldsymbol{x})$ is the Jacobian matrix defined as:

$$\boldsymbol{J}_{\boldsymbol{f}}(\boldsymbol{x}) = \frac{\partial \boldsymbol{f}(\boldsymbol{x})}{\partial \boldsymbol{x}} = \begin{pmatrix} \frac{\partial f_1}{\partial x_1} & \cdots & \frac{\partial f_1}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_m}{\partial x_1} & \cdots & \frac{\partial f_m}{\partial x_n} \end{pmatrix} = \begin{pmatrix} \nabla f_1(\boldsymbol{x})^T \\ \vdots \\ \nabla f_m(\boldsymbol{x})^T \end{pmatrix} = \left(\frac{\partial \boldsymbol{f}}{\partial x_1}, \cdots, \frac{\partial \boldsymbol{f}}{\partial x_n}\right) \in \mathbb{R}^{m \times n}. \tag{2.65}$$

Once the gradients are obtained, the gradient-descent optimization algorithms will be used for upgrading parameters. The BP algorithm can be extended to general DAGs. Thanks to the computational graph and auto differentiation techniques, the BP algorithm can be fast implemented without manual calculation, in practical network training.

A proper loss function is critical for network training, which determines the learning criteria for network training. There are various loss functions for different tasks. For example, in regression problems, the mean square error (MSE) loss (L_2 norm) is widely used, which is defined as:

$$MSE = \frac{1}{N}\sum_{i=1}^{N}\left(Y_i - \hat{Y}_i\right)^2, \tag{2.66}$$

where N is the total number of data samples (batch size), Y_i is the label, and $\widehat{Y_\iota}$ is the predicted value. Similarly, the mean absolute error (L1 norm) is defined as:

$$MAE = \frac{1}{N}\sum_{i=1}^{n}\left|Y_i - \hat{Y}_i\right|^2, \tag{2.67}$$

MAE does not penalize the outliers with squares. Sometimes, more complex Huber loss is used instead of MAE and MSE. This loss function is defined as:

$$L_\delta(Y,\hat{Y}) = \begin{cases} 1/2\left|Y_i - \hat{Y}_i\right|^2 & for\ |Y_i - \hat{Y}_i| \leq \delta \\ \delta|Y_i - \hat{Y}_i| - 1/2\delta^2, & otherwise \end{cases}, \tag{2.68}$$

where δ is a hyperparameter.

The cross-entropy loss and Kullback–Leibler divergence (KL- divergence) are common choices for classification problems. Both loss functions measure how the predicted probability distribution $\hat{Y}_i$ is similar to the target distribution Y_i. The cross-entropy loss is defined as:

$$L(Y,\hat{Y}) = -\frac{1}{N}\sum_{i=1}^{n}\left[Y_i log\hat{Y}_i + (1 - Y_i)log\left(1 - \hat{Y}_i\right)\right]. \tag{2.69}$$

The KL- divergence is defined as:

$$D(Y,\hat{Y}) = \sum Y log\frac{Y}{\hat{Y}}. \tag{2.70}$$

Of course, this section cannot exhaust all possible loss functions. Many customized loss functions are developed for specific tasks. For example, dice coefficient loss and shape-aware loss are proposed in image segmentation. In image super-resolution, kinds of perceptual loss functions are designed to measure the similarity between two images. The choice of specific loss functions should be based on analyzing particular problems [98].

2.7.2 Optimization algorithms

The optimization of ANNs is considerably complicated because ANNs are nonlinear systems, and their loss functions are high-dimensional non-convex functions. The challenges in network training can be summarized as follows:

a. For high-dimensional non-convex optimization problems, the existence of local minima and saddle points makes the optimization intractable. The optimization often stalls at local minima, and the gradients will vanish at saddle points.

b. Deep ANNs suffer from gradient vanishing and explosion, causing the failure of gradient descent algorithms.

c. There is no one-size-fits-all optimization algorithm considering the variety of neural network architectures.

d. Due to the extensive parameters of deep ANNs and large training datasets, it is impractical to use computationally expensive second or higher-order optimization algorithms to train the neural networks.

The first-order gradient descent algorithms are the mainstream for network training. The primary optimization algorithm is Stochastic Gradient Descent (SGD). Consider a neural network $\boldsymbol{f}\,(\boldsymbol{x},\ \boldsymbol{\theta})$, where $\boldsymbol{\theta}$ is the network parameters. During the training process, we uniformly sample B training data $\boldsymbol{S}_t = \{(\boldsymbol{x}^1, \boldsymbol{y}^1),\ \cdots, (\boldsymbol{x}^B, \boldsymbol{y}^B)\}$ from the training dataset each time. The derivative with respect to $\boldsymbol{\theta}$ at t^{th} iteration is:

$$\boldsymbol{g}_t(\theta) = \frac{1}{B} \sum_{(\boldsymbol{x},\boldsymbol{y}) \in \boldsymbol{S}_t} \frac{\partial \boldsymbol{L}\big(\boldsymbol{y}, \boldsymbol{f}(\boldsymbol{x}, \theta)\big)}{\partial \theta}, \tag{2.71}$$

The updated parameter is:

$$\triangle\, \theta_t = -\eta \boldsymbol{g}_t, \tag{2.72}$$

where η>0 is the learning rate $\triangle\theta_t \triangleq \theta_t - \theta_{t-1}$. We usually use minibatch SGD to improve computational and statistical efficiency. The small batch number B can fully utilize the hardware resource and significantly improve the convergence speed. Eq. (2.72) shows that the learning rate and the gradient estimation play an important role in gradient descent. Therefore, researchers propose improved algorithms based on minibatch SGD to ensure effective and efficient network training. According to different improved aspects, the improved optimization algorithms can be divided into three categories: adaptive learning rate, modifying gradient estimation, and combined methods.

The commonly used optimization algorithms with adaptive learning rates include AdaGrad, AdaDelta, and RMSprop [107]. AdaGrad adaptively adjusts the learning rate of each parameter at every iteration. We first calculate the cumulative squared gradient at t^{th} iteration:

$$G_t = \sum_{i=1}^{t} \boldsymbol{g}_i \odot \boldsymbol{g}_i, \tag{2.73}$$

where $\odot$ is the element-wise multiplication, $\boldsymbol{g}_i \in \mathbb{R}^{|\theta|}$ is the gradient at t^{th} iteration. The updated parameter is:

$$\triangle\theta_t = -\frac{\eta}{\sqrt{G_t + \varepsilon}} \odot \boldsymbol{g}_t, \tag{2.74}$$

where η is the initial learning rate, and ε is a small value (typical 10^{-8}) to keep the numerical stable. For AdaGrad, a parameter that has a large cumulative squared gradient will lead to a smaller learning rate and vice versa. The overall learning rate will decrease as the iteration epoch increases. However, this would lead to a minimum learning rate after many iterations and the optimization will stall. RMSprop can fix the early stop of optimization caused by a monotonically decreased learning rate in AdaGrad. RMSprop first calculates the decaying average of gradient updates:

$$G_t = \beta G_{t-1} + (1-\beta)\boldsymbol{g}_t \odot \boldsymbol{g}_t, \tag{2.75}$$

where β is the decay rate. The updated parameter is the same as Eq. (2.75). AdaDelta is a further extension of AdaGrad by adapting learning rates based on the decaying average of gradient updates, instead of accumulating all past gradients. It is similar to RMSprop.

A typical algorithm for optimization algorithms with modifying gradient estimation is the Momentum method. Momentum is a physics concept, meaning an object keeps its original moving tendency. The momentum method mimics this physics concept and uses the accumulated gradient to replace the current gradient. The iterative gradient can be considered as acceleration. The updated parameter at t^{th} iteration is:

$$\triangle \theta_t = \rho \triangle \theta_{t-1} - \alpha \boldsymbol{g}_t, \tag{2.76}$$

where ρ is the momentum factor and is typically set as 0.9. α is the learning rate.

The combined method combines both the adaptive learning rate and the Momentum method. For example, The Adaptive Moment Estimation Algorithm (Adam) combines the RMSprop and Momentum methods.

2.7.3 Regularizations

Deep neural networks have a strong fitting ability to train data due to their high complexity and nonlinearity. However, it could cause overfitting problems, i.e., the excellent performance on training datasets cannot be generalized to other testing datasets. Regularization is a family of various techniques that can restrict network complexity and avoid overfitting. Commonly used regularizations include introducing constraints, dropout, and early stopping.

The network optimization with l_1 or l_2 constraints can be described as:

$$\boldsymbol{\theta}^* = arg \min_{\boldsymbol{\theta}} \frac{1}{N} \sum_{n=1}^{N} \boldsymbol{\mathcal{L}}\big(y^n, \boldsymbol{f}(\boldsymbol{x}^n; \boldsymbol{\theta})\big) + \lambda \ell_{1\,or\,2}(\boldsymbol{\theta}), \tag{2.77}$$

where $\boldsymbol{\mathcal{L}}(\cdot)$ is the loss function, $\boldsymbol{f}(\boldsymbol{\theta})$ the neural network with parameter set $\boldsymbol{\theta}$, $\ell_{1\,or\,2}$ the 1- or 2-norm and λ the regularization coefficient. ℓ_1 and ℓ_2 can be added to the regularization terms. It is known as elastic net regularization.

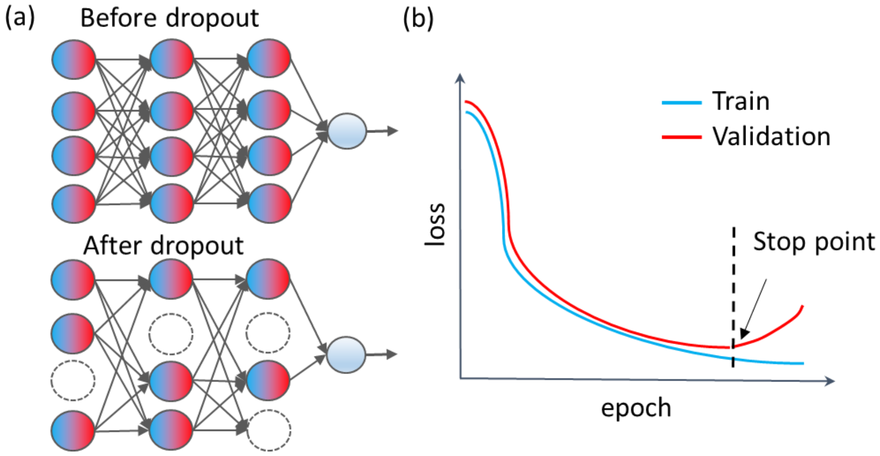


Fig. 2.24. Drop out and early stopping techniques to prevent overfitting.

The dropout method randomly drops out some neurons during training. It can also drop out the input neurons, equivalent to adding noise to the data. Fig. 2.24 (a) shows the FNN before and after dropout. The dropout method can efficiently improve the network's robustness and avoid overfitting. Another simple but effective regularization method to prevent overfitting is the early stopping method. Fig. 2.24 (b) illustrates the principle of early stopping. During the training, we use an independent validation set to validate the training. The iteration terminates once the validation loss stops decreasing.

In summary, sections 2.5-2.7 systematically review the fundamental knowledge of ANNs. It should be emphasized that deep learning is a highly empirical discipline at present. To efficiently train ANNs, researchers conclude many tricks and techniques learned from extensive trial and error. These methods include parameter initialization, data preparation, data argumentation, and hyperparameter optimization. In addition, the graphic neural network and reinforcement learning are not presented because they are out of the scope of this thesis.

2.8 Deep learning in microscopy

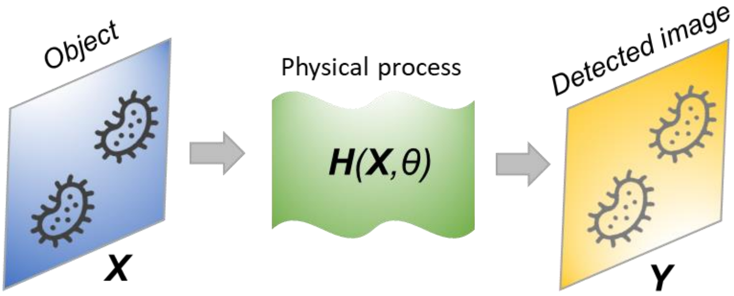


Fig.2.25. General image formation process.

DL has revolutionized optical imaging by introducing a new data-driven approach. Fig. 2.25 illustrates the image formation process of general imaging modalities. The object modulates the propagating light. After undergoing a specific physical process (e.g. light propagation through an optical path), the modulated light carrying the object's information is detected by an image sensor. This process can be described as:

$$\boldsymbol{Y} = \boldsymbol{H}(\boldsymbol{X}, \theta), \tag{2.78}$$

where $\boldsymbol{X}$ is the object image, $\boldsymbol{H}$ is the transfer function of the physical process, θ is other effect factors, and $\boldsymbol{Y}$ is the detected image. For example, the optical system can be characterized by the point spread function (PSF) in microscopy systems. The PSF describes the impulse response of the optical system to a point light source. The Fourier transform of the PSF is the Optical Transfer Function (OTF). The mode and phase of the OTF are the Modulation Transfer Function (MTF) and the Phase Transfer Function (PTF), which describe the system characterizations in space and frequency domains, respectively. The PSF can represent $\boldsymbol{H}$, and the detected image $\boldsymbol{Y}$ is:

$$\boldsymbol{Y} = PSF * \boldsymbol{X} \tag{2.79}$$

The PSF should be evaluated first, and then $\boldsymbol{X}$ can be recovered by deconvolving the PSF with the detected image $\boldsymbol{Y}$. However, in more advanced imaging systems such as optical coherence tomography, digital holographic imaging, Fourier lamination imaging, and scattering medium imaging, $\boldsymbol{H}$ is inexplicit and has no analytical formulation. To reconstruct $\boldsymbol{X}$, it should introduce many approximations and assumptions to simplify the model of the image formation process. Then $\boldsymbol{X}$ is obtained by solving the nonlinear inverse problem using iterative optimization algorithms. Such reconstruction methods are only suitable for some specific scenarios, and the inverse problems often become severely ill-posed.

DL breaks this "physical-model-driven" paradigm. Due to DL's excellent learning ability from data, it can directly map detected image $\boldsymbol{Y}$ to the object image $\boldsymbol{X}$, thus avoiding the difficulty of developing a forward model and solving the inverse problems. DL has achieved huge success in a wide range of biomedical optics imaging techniques within just a few years. DL-enhanced microscopy significantly improves imaging performance beyond instrumental limitations. It even develops novel functionalities for better image interpretation. DL also provides a novel route to analyze FLIM images. The recent progress is summarized as follows:

2.8.1 Imaging performance improvement

SNR. Obtaining sufficient SNR is critical for better observation of samples. However, it usually requires high light exposure or a long imaging time, which can lead to photo-bleaching, photo-toxicity, and slow imaging speeds. DL techniques provide a neat way to circumvent this dilemma by training a neural network to map the noisy images to the corresponding noise-free images. For example, Weigert et al. developed a DL algorithm using U-net as a backbone to achieve SNR enhancement in fluorescence microscopy [108]. The network was trained using image pairs with low and high light exposures, and then it directly converted noisy input images to high-SNR output images. This study demonstrated that DL could effectively recognize noise and reconstruct high-SNR images using 60-fold fewer photons. Similarly, DL techniques also find applications for SNR-enhancement tasks in other microscopy modalities such as widefield, confocal, multi-photon, and light-sheet microscopes [109-112].

Resolution. Optical microscopes often face a trade-off between spatial resolution and field-of-view (FOV) or depth-of-field (DOF) due to the need for oil immersion objectives or large magnification objectives. DL techniques have emerged as a solution to this problem by using neural networks to enhance the spatial resolution of low-resolution images. In 2017, Rivenson proposed a DL algorithm to improve the spatial resolution of optical microscopy [113]. To train the neural network, a microscopic imaging dataset with low- and high-resolution image pairs is prepared using a 40x and 100x oil immersion objective lens, respectively. Then, a CNN learns the mapping relationship between the image pair. Once the network is well trained by massive data, it will learn the high-frequency information of images and directly transform low-resolution images obtained from a 40x objective lens to the corresponding high-resolution images quickly. As a result, the microscopy system can simultaneously obtain high spatial resolution and large FOV. In addition, the imaging process is substantially simplified because oil immersion is no longer required.

Similarly, GANs are also used for spatial resolution enhancement in microscopy systems. Wang et al. proposed a GAN-based super-resolution algorithm for reconstructing a high-resolution image from low-resolution measurements, obtaining an enlarged FOV and extended DOF simultaneously [114]. The DL-based resolution enhancement techniques are not limited to optical microscopies. They are also widely applied to electron, fluorescence, and holographic microscopy [115-117]. Currently, DL techniques mainly focus on the aspect of image processing. Further research should combine the design of both optical imaging systems and neural networks to reconstruct object information rapidly and efficiently.

Reconstruction speed. For many kinds of microscopy modalities, image formation requires a large number of measurements and complex post-processing to reconstruct the image. DL techniques can significantly improve imaging acquisition and reconstruction speeds by reducing measurement times and bypassing post-processing. Photo-activated localization microscopy (PALM) and stochastic optical reconstruction microscopy (STORM) are examples of improving image reconstruction speed using DL. Both PALM and STORM are widefield super-resolution microscopies that can obtain images with a spatial resolution beyond the diffraction limit. They use specific fluorescent molecular markers and need thousands or even millions of acquisition times to form the image. Therefore, these methods are incredibly time-consuming and rely on complex and expensive hardware systems. The advent of DL offers a new strategy to accelerate the reconstruction speed of STORM and PALM. It has been reported that super-resolution images can be reconstructed from up to two orders of magnitude fewer frames than conventional imaging processes, significantly reducing the imaging acquisition time [117-119]. Another example is phase imaging reconstruction. Only about 3% of data is required to reconstruct gigapixel-scale phase images compared to the conventional sequential acquisition scheme through DL [109].

2.8.2 Novel functionalities

Cross-modality image transformation. The typical applications of cross-modality image transformation are virtual staining and labeling. This histological analysis of tissue samples plays a fundamental role in clinical disease diagnosis. However, the preparation of tissue samples is typically laborious, lengthy, and expensive. Before being imaged by microscopy, the samples should be formalin-fixed and paraffin-embedded (FFPE), sectioned into 2~10 μm thin slices, stained, mounted, and sealed on a glass slide. DL brings a virtual stain technique to circumvent these tedious steps, substantially simplifying the staining procedure and reducing costs. The main idea of virtual stain is to train a CNN using image pairs of unstained and histologically stained images. CNN can learn the hidden and complex relationships between massive training image pairs and then converts the unstained input images to target stained images. Research has demonstrated that the virtually stained images show superior performance, which has passed a rigorous examination by board-certified pathologists [120]. The unstained input images could be autofluorescence, phase, and fluorescence lifetime images. The targeted stained images could be haematoxylin and eosin (H&E), Masson's Trichrome and Jones' stain. Similarly, DL can also transform label-free contrasts such as brightfield, phase, and polarization images into fluorescence-labeled images [121-124]. Conventional physical fluorescent labels suffer from a lengthy procedure, the requirement of specialized reagents, inconsistency, a limited number of labels, and inevitable perturbations. Virtual labeling techniques based on DL can overcome these drawbacks, fast and accurately labeling multiple subcellular constituents such as DNA, nucleoli, mitochondria, cell membrane, and cell states.

Classification and segmentation. The powerful ability of DL to capture high-level image features can be used for image classification and segmentation, assisting researchers in better understanding and interpreting the experimental results. DL-based classification has been used for cancer screening, classifying disease progression, and identifying cell types and states. For example, the DL combined with spatial light interference microscopy has been used for classifying cancer and benign colorectal tissues. The accuracy of the test dataset reaches 99% [125]. The combination of DL and Deep-UV excitation fluorescence microscopy has been demonstrated to classify lymph node metastasis [126] efficiently. A recent study shows that applying DL in imaging flow cytometry can be used for reconstructing cell cycle progression [127]. Results show that DL can precisely classify the cell states into seven categories, including G1/S/G2, prophase, metaphase, and telophase. The accuracy for the G1/S/G2 phase can reach 98.73% ± 0.16%.

DL-based segmentation is also a powerful tool for understanding cellular information in depth. The automatic segmentation of cells in *ex vivo* tissue enables researchers to mine more cell information, such as cell morphology, cell frequency, and spatial distribution of sub-cellular populations. The DL segmentation techniques using U-Net and its derivatives as backbones have been applied to semantic segmentation tasks. For example, they have been used for full-organ, nuclear, and reflectance confocal microscopy mosaic segmentation tasks [128]. In addition, DL-based instance segmentation has been successfully applied to cell segmentation for object detection [129].

2.8.3 DL-enhanced FLIM analysis

As discussed in section 2.4, conventional FLIM analysis relies on model-based fitting methods such as LSF and MLE or model-free non-fitting methods such as phasor analysis and CMM. The fitting methods are computation-intensive and time-consuming, unsuitable for fast and real-time FLIM analysis. Phasor analysis and CMM are more rapid as they do not need iterative optimization. However, they only deliver accurate results with sufficient photons (large SNR) and require careful calibration. Recently, DL techniques opened the possibility for fast and robust FLIM analysis without resorting to data fitting. In 2016, our research group first applied DL in fast FLIM analysis. Wu et al. proposed an MLP-based deep learning algorithm, which shows a 180-fold faster speed than conventional LSF [130]. However, this algorithm suffers from disadvantages. First, it follows the "tail-fitting" approach and does not consider the IRF. Second, due to MLP's limited learning ability, the proposed algorithm cannot resolve multi-exponential decay models. One feasible method to improve the learning ability of MLP is to use the phasor coordinates of fluorescence decay as network input. The phasor coordinates can be viewed as a simple feature engineering [131]. The network can better learn the features of input data and can also significantly reduce the training time. Later, more advanced DL algorithms were developed to address the above disadvantages. In 2019, Smith et al. proposed a 3D-CNN-based FLIM algorithm, FLI-Net, for fast analyzing the bi-exponential decay model [132]. The input of FLI-Net is the 3D spatial-temporal data cube, and the output is bi-exponential decay parameters. FLI-Net is trained using model-generated synthetic data without adding an extra experimental burden. Then the well-trained algorithm was validated by various biological applications, such as metabolic analysis and FLIM-FRET. FLI-Net outperforms conventional FLIM algorithms in terms of accuracy and speed, especially in low-photon conditions. In addition, the GAN-based FLIM algorithm, flimGANE, was also proposed recently [133]. However, these algorithms still suffer low efficiency, resulting in large hardware resource consumption and prolonged training time. Despite some disadvantages, it is no doubt that DL techniques have revolutionized FLIM analysis and opened the possibility of more novel applications.

Chapter 3

Pixel-wise Deep Learning Algorithm for Fast Fluorescence Lifetime Imaging Analysis

3.1 Introduction

Data processing is a critical aspect of FLIM techniques because FLIM is an indirect imaging modality, and the fluorescence lifetime should be extracted from the measured decay histogram. Conventionally, the dominant methods for fluorescence lifetime estimation are curve-fitting-based algorithms such as LSF and MLE. Pre-defined single or multi-exponential decay models fit the measured decay histogram in each pixel to estimate the fluorescence lifetime. The main drawbacks of curve-fitting-based algorithms are as follows:

1. The fitting processes rely on iterative-based optimization, which is computationally intensive and time-consuming.
2. Prior knowledge of the lifetime parameters is required for setting proper initial conditions.
3. Data analysis needs specialized software, which is only suitable for offline analysis.

The advent of DL provides us with a new data-driven approach to addressing the above challenges. DL features hierarchical representation learning, showing a powerful ability to learn complex functions and high-dimensional data, and outshines conventional machine-learning techniques [85]. It opens a door for fitting-free fast lifetime calculation in FLIM applications. Our group first adopt the DL technique to analyze FLIM data by developing an MLP-based dense neural network algorithm [130]. DNN shows a remarkable advantage in fluorescence lifetime estimation regarding accuracy and calculation speed. However, it cannot analyze multi-exponential decay models due to DNN's inadequate learning ability. Later, a more complex 3D CNN algorithm, FLI-Net, was proposed to rapidly analyze FLIM data with multi-exponential decays [132]. Since CNNs are mainly designed for computer vision tasks, their architectures tend to be 2D, 3D, or even higher dimensions [133-135]. FLI-Net also follows this principle: fluorescence data are treated as a 3D image with two spatial and one temporal dimension. Consequently, FLI-Net has a large model size, prolonged training time, and slow inference speed. In addition, high-dimensional neural networks like FLI-Net usually require extensive hardware resources and large power consumption, making them difficult to implement on embedded systems.

This chapter develops a light-weighted and hardware-friendly pixel-wise DL algorithm to address the challenges above. We show that our pixel-wise DL algorithm has fewer parameters, short training time, fast inference speed, and a powerful ability to resolve multi-exponential decays accurately. The key innovation of our proposed algorithm lies in adopting one-dimensional (1D) CNN architectures [136]. 1D CNNs use 1D convolutional kernels in convolutional layers. They have been widely used for real-time electrocardiogram monitoring and structural damage detection [136-138]. We demonstrate that the 1D CNN-based pixel-wise DL algorithm is more suitable for FLIM analysis, possessing unparalleled advantages in many aspects. The main reasons are summarized below:

1. For FLIM data analysis, only temporal decay features need to be evaluated without considering spatial positions during the data analysis. In other words, FLIM data should be treated as a 1D time series instead of a 3D matrix. It is straightforward because the lifetime calculation process is to fit decay curves pixel-by-pixel in both point-scan and wide-field FLIM measurements. In some cases, spatial binning is required to improve the SNR, but the fitting process works similarly.

2. Applying high-dimensional CNNs increases the training and calculation complexity significantly and requires hardware with much more computing power and larger memory. In contrast, 1D CNNs have fewer parameters and consume minimal hardware resources.

3. 1D CNNs are much faster, as the float-point additions and scalar multiplications replace the complex matrix operations from high dimensional convolution in 1D CNNs. Thus, it is promising for real-time FLIM applications. The design and training of the neural network have also become much easier.

4. Implementing 1D CNNs on electronic hardware such as FPGA devices is easier, only requiring simple additions and multiplications [139-141]. Simpler operations also guarantee simpler architectures and lower hardware resources needed. Therefore, they are more hardware-friendly and suitable for cheap, portable devices with limited CPU computing power, memory, and power supplies. Besides, simple 1D CNN architectures allow the processing of multiple parallel series simultaneously, providing a fast and feasible way for processing FLIM images. They can work as embedded processors compatible with existing FLIM systems.

3.2 Training Data Preparation

For the development of data-driven algorithms, the acquisition of a large amount of training data is critical. In addition, the training data directly affect the performance of the trained algorithms. One practical method is to generate synthetic decay curves with known parameters as the training datasets using Monte Carlo simulation because the mathematical models of the fluorescence decay curves and noise characteristics have been well developed. Under a sufficient photon-count condition, the synthetic measured decay is the multi-exponential decay mode convolved with the IRF, and the noise can be described as an addictive noise term:

$$Y[t] = K\left[\sum_{i=1}^{n} \alpha_i\, e^{-t/\tau_i}\right] * IRF[t] + \epsilon[t], t = 1, \ldots, N \tag{3.1}$$

where the fraction ratio α_i (i=1, …, n) obeys a uniform marginal distribution in the range from 0 to 1 with a fixed sum of 1. The convolution result of the decay curve and $IRF(t)$ was normalized to 1 and then multiplied by the peak intensity K *(maximum photon counts).* $\varepsilon(t)$ is the *Poisson* noise. For larger photon counts (generally total photon counts > 20), $\varepsilon(t)$ is approximated by a normal distribution and expressed as:

$$\epsilon[t] = \sqrt{K\left[\sum_{i=1}^{n} \alpha_i \, e^{-t/\tau_i}\right] * IRF[t]} \cdot \mathcal{N}(0,1) \tag{3.2}$$

where $\mathcal{N}(0,1)$ represents a normal distribution with zero mean and a standard deviation of 1., the peak intensity K can be set as follows to generate a specified SNR:

$$K = \mathrm{SNR}^2 / \sum_{\mathrm{t}=1}^{\mathrm{N}} \left[\left[\sum_{\mathrm{i}=1}^{\mathrm{n}} \alpha_{\mathrm{i}} \, \mathrm{e}^{-\mathrm{t}/\tau_{\mathrm{i}}}\right] * IRF[t]\right] \tag{3.3}$$

Our study uses homemade code in MATLAB (MathWorks Inc., Massachusetts, USA) to generate the synthetic FLIM datasets. The simulation parameters are specified according to the experiments' TCSPC (SPC-830, Becker & Hickl GmbH, Berlin, Germany). The IRF is fitted with a Gaussian function and expressed as $IRF[t] = e^{[-(t-t_0)^2 \cdot 4ln2h^2/FWHM^2]}$, where t_0 is the peak position. Table 3.1 shows the detailed parameter setting:

Table 3.1 Parameters for Generating Training Datasets

Number of time bins (N)	256
Bin width (h) [ns]	0.039
Total observation period (T) [ns]	10
The peak position of IRF (t_0)	U[13, 15]
FWHM of IRF [ps]	167.3

In the above table, U[•] denotes a uniform distribution. We use the random shift of t_0 to simulate the laser jitter.

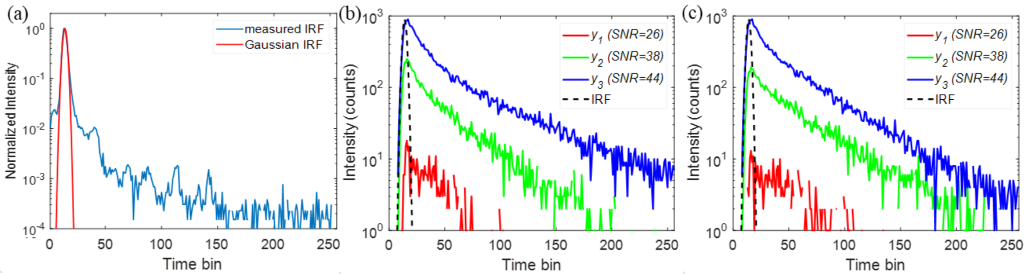


Fig. 3.1. (a) The measured IRF and the fitted IRF using the Gaussian function. (b) Examples of simulated bi-exponential decay curves with different SNRs. The parameters (τ_1, τ_2, α, SNR) of $y1$, $y2$, and $y3$ are (0.1, 2, 0.3, 26), (0.3, 1.5, 0.6, 38), and (0.5, 1.5, 0.8, 44), respectively. The dashed line represents the Gaussian IRF. (c) Examples of simulated tri-exponential decay curves with different peak intensities. The parameters (τ_1, τ_2, τ_3, α_1, α_2, SNR) of $y1$, $y2$, and $y3$ are (0.1, 0.9, 1.8, 0.2, 0.3, 26), (0.2, 1.1, 1.9, 0.4, 0.2, 38), and (0.3, 1.2, 2.0, 0.6, 0.2, 44), respectively. The dashed line represents the Gaussian IRF.

Fig. 3.1 (a) shows the synthetic and experimental IRF profiles for our FLIM system. The tail in the experimental IRF has a very low intensity and therefore can be neglected. Fig. 3.1 (b) and (c) show the synthetic bi- and tri-exponential decays with variant SNRs.

3.3 Algorithm Development

3.3.1 Neural Network Design

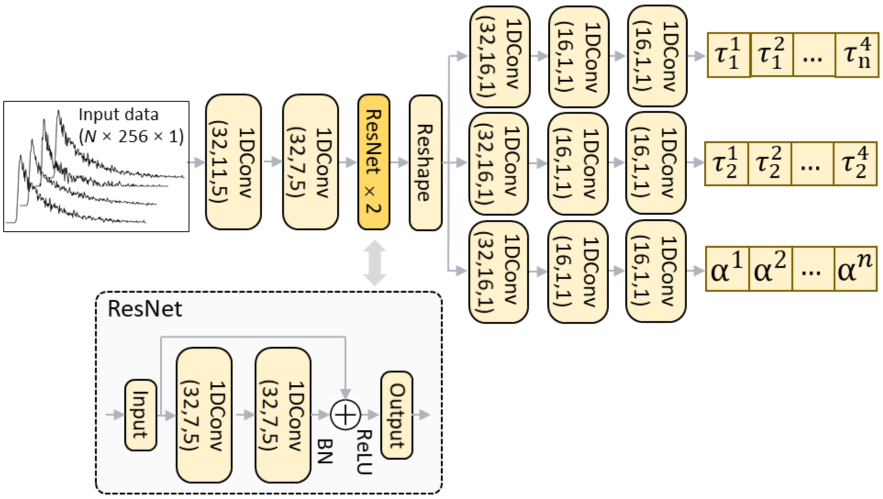


Fig. 3.2. The architecture of the proposed pixel-wise DL algorithm for bi-exponential analysis. The detailed hyperparameters in each layer in parenthesis represent the neuron number, the kernel size, and the stride size, respectively. The input data is an N×256×1 time series, where N is an arbitrary integer. The outputs are three N×1 arrays. The dotted box shows the architecture of the ResNet block.

This section presents a 1D CNN-based pixel-wise DL algorithm. The "pixel-wise" denotes that the algorithm processes each independent pixel instead of the whole image. Fig. 3.2 shows the neural network architecture of the proposed pixel-wise DL algorithm. For simplification, we use CNN to indicate the pixel-wise DL algorithm hereafter. The neural network is composed of 1D convolutional blocks. In each block, a 1D convolution layer, a batch normalization (BN) layer, and a nonlinear activation layer are connected in sequence. The network is specially designed for bi-exponential fluorescence decay analysis because the bi-exponential decay model is widely used in FLIM analysis, like FLIM-FRET applications. It consists of two parts: the main branch for extracting the high-dimensional abstract features of the input decays and the sub-branches for reconstructing different parameters. For the network design, the first two convolutional blocks have a wider sliding window (larger kernel size) and a large stride, so they are expected to capture more general features of the decay curves. Two residual neural networks (ResNet) follow the convolutional blocks, through which the extracted features are learned using residual learning. A ResNet is a network connection that adds a shortcut to reinject the past output to a later output [135]. It aims to tackle two common problems: 1) the vanishing gradients and 2) representational bottlenecks during network training. Three output branches are designated for extracting three different lifetime parameters for bi-exponential decays (the short lifetime τ_1, the long lifetime τ_2, and the fraction ratio α), which are composed of three 1D CNN layers with pointwise convolution (kernel size = 1) and a decreasing neuron size. Pointwise convolution is used to implement feature pooling, thus effectively reconstructing the lifetime parameters in the final outputs. The rectified linear units (ReLU) function is used in nonlinear activation layers. The hyperparameters of the network such as the feature size, convolution kernel, and stride of convolutional layers, were optimized using the exhaustive searching method, labeled in Fig. 3.2 in detail [142].

3.3.2 Neural Network training

The proposed neural network was developed using the Pytorch library in Python 3.8. The loss function of our designed neural network *L* is:

$$L(\Theta) = \frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{M}\left|\left|H_j(Y^i, \Theta) - \widehat{Y}_j^i\right|\right|_2^2, \tag{3.4}$$

where *H* is the j^{th} (j=1, …, M) end-to-end mapping function and *M* is the number of the output branches (M = 3 for bi-exponential decays). *Y* is the input signal and $\widehat{Y}_j$ is the corresponding ground-truth target of the j^{th} output branch. *N* is the batch size number, and *Θ* is the hyperparameter of the network. The optimizer used the RMSprop algorithm with a learning rate of 1e-5. The batch size is 128, and the training epoch is 500. An early stopping regularization with 25 patient epochs (5% of total epochs) is applied to avoid overfitting.

3.3.3 The Advantages of Pixel-wise DL algorithm for FLIM analysis

Apart from the advantages of fast speed, easy implementation, and low resource consumption mentioned above, the pixel-wise DL algorithm is more flexible for analyzing fluorescence images. Unlike higher dimensional neural networks that need to process whole images simultaneously, 1D CNN can sort valid pixels by setting a threshold or regions of interest. Meanwhile, multiple parallel series processing capabilities of 1D CNNs allow arbitrary pixels to be processed simultaneously. Therefore, the 1D CNN has a better calculation efficiency, especially for FLIM images containing a large background area with no useful information.

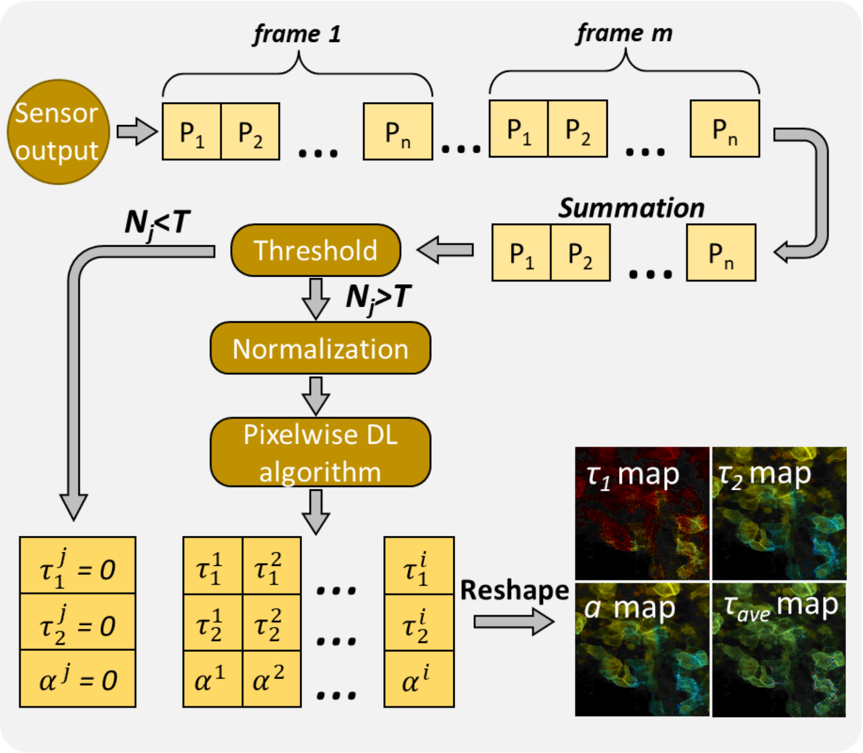


Fig. 3.3. Workflow of 1D-ConvResNet for calculating 2D FLIM images obtained from a sensor. The data readout of the sensor follows the pipeline principle. The raw output data flow is arranged pixel-by-pixel and frame-by-frame. P_i refers to the pixel in the sensor. 1D CNN can directly work as an embedded processor to process the data.

In addition, the data readout of sensors such as PMTs and SPADs in FLIM systems usually follows a pipeline principle, i.e. the measured raw data is sorted pixel-by-pixel and frame-by-frame. 1D CNNs can directly process the output data without extra data conversion. Fig.3.3 shows the image processing procedure. The embedded CNN processor can focus on the pixels with an intensity above the threshold. Those below the threshold are considered background pixels, and their lifetimes are ignored.

3.4 Algorithm Evaluation

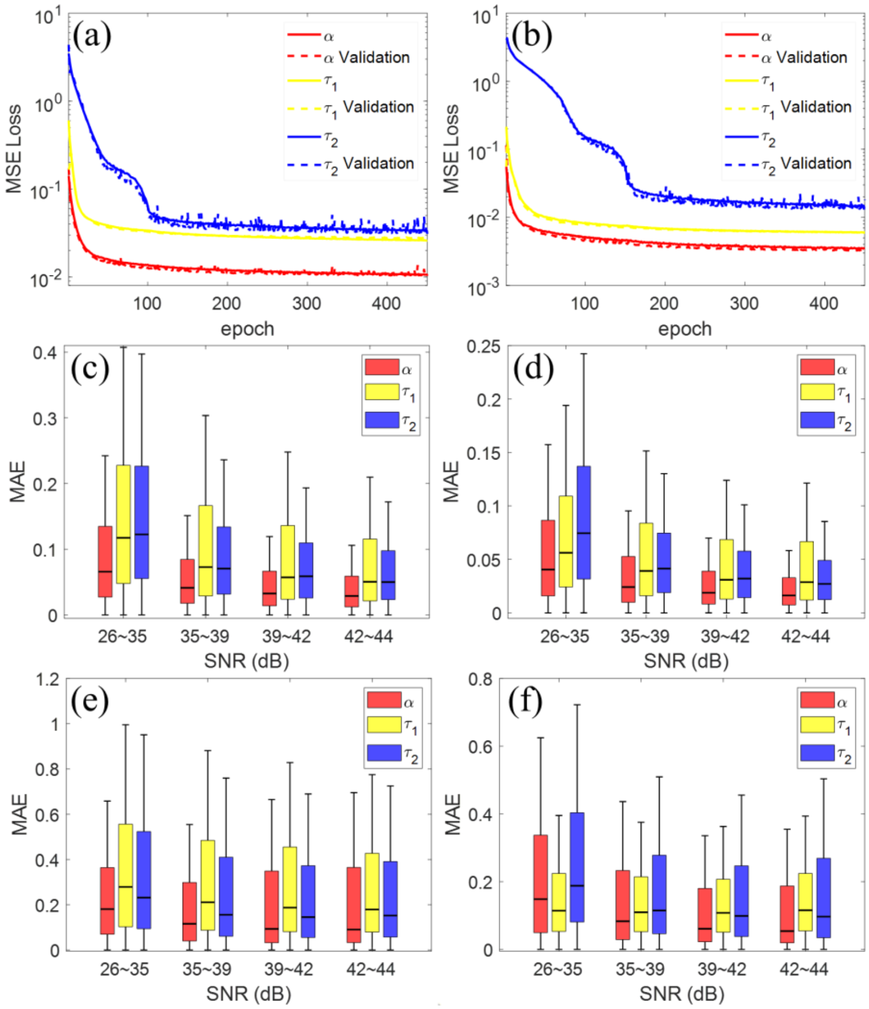


Fig. 3.4. The mean absolute error (MAE) of the training and validation for τ₁, τ₂, and α using training Datasets A (a) and B (e). The corresponding MAE of predicted results for the three parameters when the network is trained by the Datasets A ((b)-(d)) and B((f)-(h)).

Two sample datasets, *Datasets A* and *B* were generated for network training. Both datasets contain 40,000 decay samples, in which 20% of the samples were used as the validation subset. In *Dataset A*, τ_1 ~ U[0.1, 1] ns and τ_2 ~ U[1, 3] ns. In *Dataset B*, τ_1 has a relatively narrower range as τ_1 ~U[0.1, 0.5] ns, whereas τ_2 remains unchanged. The SNR of the training datasets follows a uniform distribution as SNR ~ U[26,44] dB, covering a wider range of noise levels. Here SNR (dB) = 20log(SNR). Fig. 3.4 (a) and (b) show the mean square error (MSE) loss of the training and validation data changing with the training epochs for *Datasets A* and *B*, respectively. All three parameters converged at a very low loss level. Compared with Fig. 3.4 (a) and (b), the loss of τ_1 is relatively large if a dataset with a wider lifetime range trains it. As a result, the predictions of τ_1 would have a large deviation. Fig. 3.4 (c) shows the boxplot of the mean average error (MAE) of the predicted results of the testing datasets using the trained network as shown in Fig. 3.4 (a). The testing dataset contains four datasets of samples with SNR ranging from 26 to 44 dB. Each dataset contains 4000 new decay curves. The lifetime range was kept the same as the training dataset. In Fig. 3.4 (c), the mean MAEs of all three lifetime parameters tend to decrease significantly as SNR increases, indicating more precise results. A noisier dataset (with a smaller SNR) has wider error distributions.

One approach is to narrow the lifetime range of interest to improve the lifetime extraction precision. It is applicable if we have prior knowledge of the lifetime distribution. As demonstrated in Fig. 3.4 (d), when τ_1 is restricted to 0.1~0.5 ns, the median MAE of τ_1 and the corresponding error distribution range have much smaller values than Fig. 3.4 (c). Meanwhile, the MAE distributions for τ_2 and α are also confined within a smaller range for the datasets with the SNR from 26 to 44dB. Further decreasing the lifetime ranges for τ_1 and τ_2 can obtain a lower MAE, showing that a narrower resolvable lifetime range could lead to higher prediction precision for all three parameters. For a better comparison, the lifetime parameters of testing datasets in Fig. 3.4 (c) and (d) were also resolved using LSF. LSF is based on the Trust Region Reflective algorithm to solve a bound-constrained optimization problem (*lsqcurvefit* in MATLAB). The constrained bounds for τ_1, τ_2, and α were set the same with their ranges in the training datasets. Figs. 3.4 (e) and (f) show the corresponding results for the testing datasets in Fig. 3.4 (c) and (d), respectively. In the whole SNR range, the performance of LSF is much worse than that of CNN. The performance of CNN for all lifetime parameters is enhanced as the SNR increases. However, the performance of LSF has no significant change when SNR varies from 35 to 44 dB. This further demonstrates the robustness and accuracy of CNN. It is worth noting that the proposed neural network has an outstanding ability to resolve the factional ratio α for a wide lifetime range. α plays an important role in evaluating the energy transfer efficiency in FERT applications. In contrast, traditional LSF or dense neural networks usually fail to extract this parameter [130].

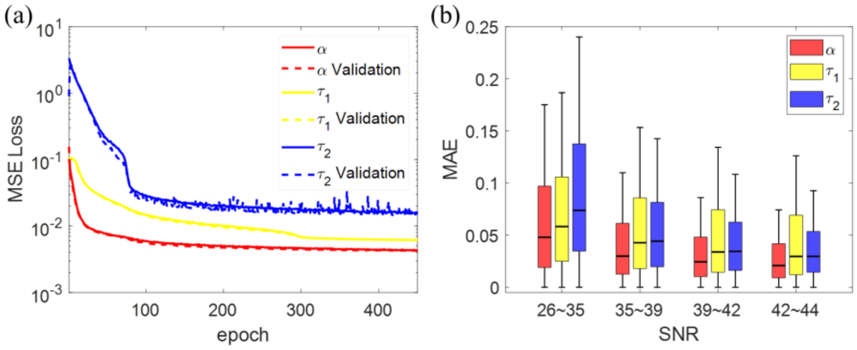


Fig. 3.5. (a) MSE loss of the training and validation for τ_1, τ_2, and α using the training dataset with τ_1 and τ_2 in the range of 0.1~ 0.5 and 1~3 ns. (b) Predicted results using the test dataset for τ_1, τ_2, and α, respectively. The lifetime parameters are τ_1 ~U[0.1, 0.5] ns, τ_2 ~U[1, 3] ns, and α~U[0, 1].

For the CNN algorithm, no deconvolution is needed because the effect of the IRF on lifetime estimations is automatically considered. Although only one IRF is considered in the training data, the CNN algorithm shows a robust performance for IRF with a wide range of FWHM. Fig. 3.5 further investigates how the FWHM of the IRF affects the training and prediction of the neural network. A new dataset was generated with the same parameters as Dataset *B*. But the FWHM of the convolved IRF in each decay sample was randomly selected from 100~300 ps. This FWHM range is typical in most laboratory FLIM systems. In Fig. 3.5 (a), the MAE curves of all three parameters converge at slightly higher values than those in Fig. 3.4. But the predicted results for test datasets are nearly the same as Fig. 3.4. It shows that our algorithm is robust for varied IRFs.

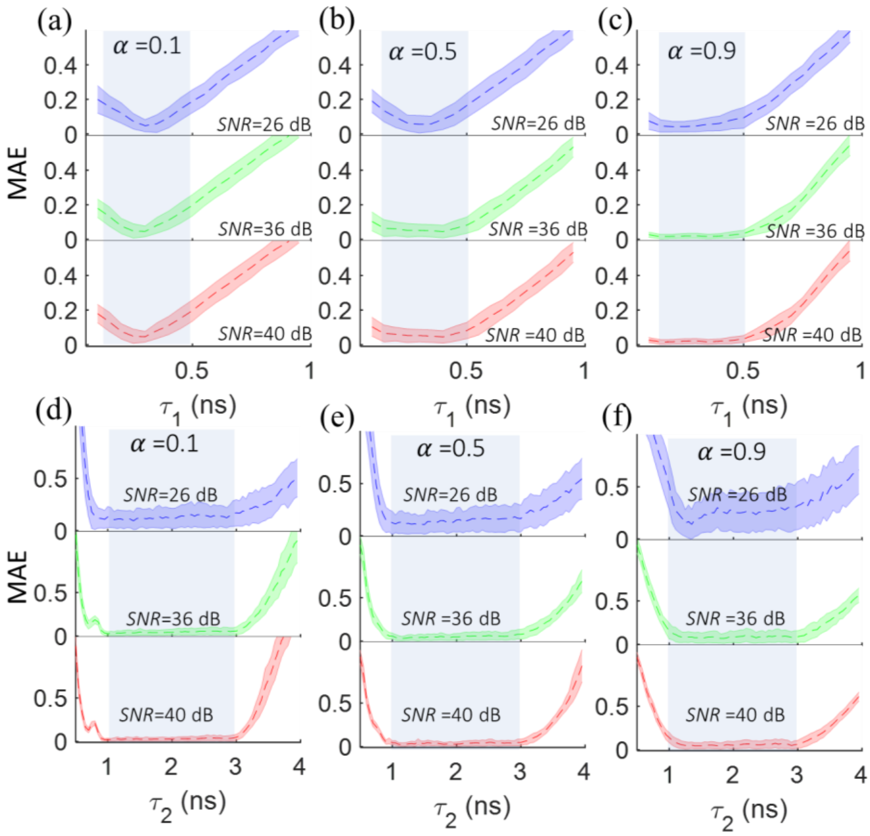


Fig. 3.6. The mean and deviation of MAE for τ_1 (a)-(c) and τ_2 (d)-(f) under different conditions. SNR = 26, 36, and 40 dB; α takes the values of 0.1,0.5, and 0.9. In the analysis, τ_2 is fixed at 2.5 ns in (a) – (c), and τ_1 is fixed at 0.3 ns in (d) – (f). The dashed areas denote the lifetime ranges of training data. In each subplot, the mean and standard deviations of the MAE of 50 random-generated decays are represented by a dashed line and shaded area, respectively.

The quantitative analysis of the MAE for τ_1 and τ_2 under different conditions was carried out to understand further the CNN algorithm's lifetime resolving capability. The neural network was trained using *Dataset B*. The new testing data have wider lifetime ranges with τ_1 and τ_2 in the range of 0.1-1ns and 1-4 ns, respectively, to investigate the lifetime resolving performance beyond the scope of the training data. Fig. 3.6 shows the results for SNR ranging from 20 to 44dB and α ranging from 0.1 to 0.9. For both τ_1 and τ_2, MAE has larger mean and standard deviation values for a smaller SNR, which agrees with the previous analysis. The resolving capability of the neural network highly depends on the training data. The neural network has a homogeneous resolving capability for the lifetimes within the range of the training data. However, the MAE of predicted results increases linearly when the lifetime exceeds the range of train data. Therefore, the resolving capability of the neural network is tunable according to the chosen training data. Also, *α* can affect the predicted results of τ_1 and τ_2. When one lifetime component has a predominant weight (for example, $\alpha \sim 1$ for τ_1), it has better precision, while another has poor accuracy.

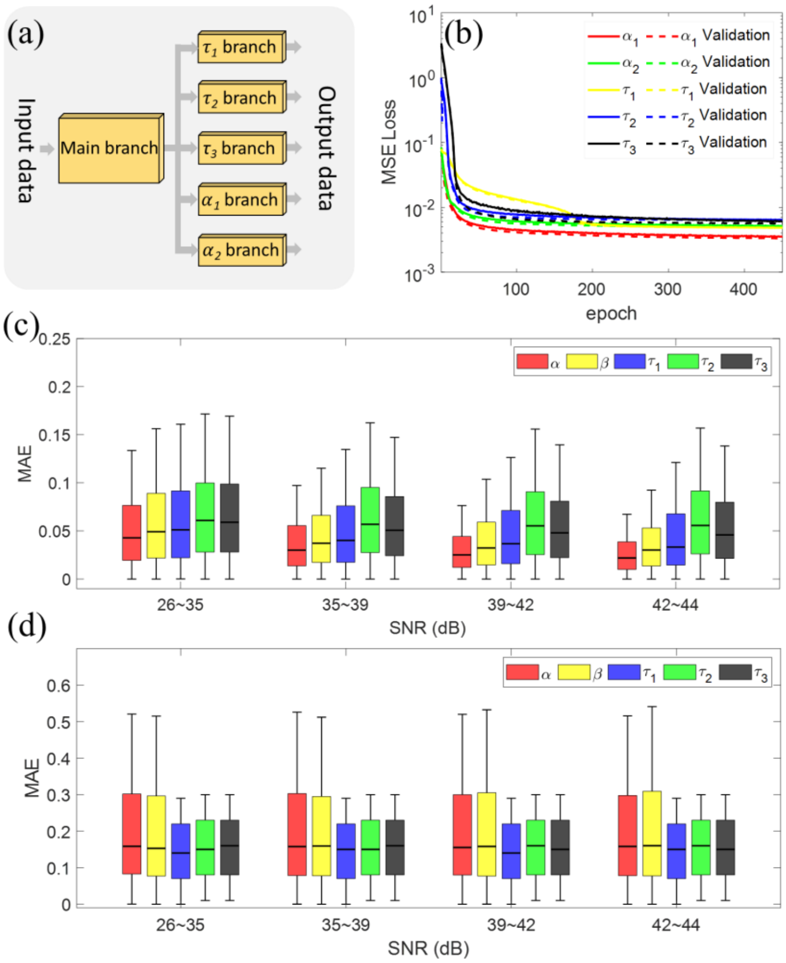


Fig. 3.7. The mean absolute error (MAE) of the training and validation for τ_1, τ_2, and α using training Datasets A (a) and B (e). The corresponding MAE of predicted results for the three parameters when the network is trained by the Datasets A ((b)-(d)) and B((f)-(h)).

The pixel-wise DL algorithm can be easily modified for *p*-exponential decay models (*p* is the number of lifetime components) by having $2p - 1$ sub-branches without redesigning the CNN architecture. Fig. 3.7 shows an example for resolving a tri-exponential decay, i.e., $p = 3$, with three different lifetime components τ_i (i =1, 2, 3), two independent fraction ratio α_i (i =1, 2), and $\alpha_3 = 1 - \alpha_1 - \alpha_2$. As shown in Fig. 3.7 (a), each parameter is assigned to a branch. The hyperparameters of the main branch and sub-branches remain the same as in Fig. 3.2. The training dataset contains 40,000 different samples. The parameter ranges of the training dataset are: τ_1~U[0.1, 0.4], τ_2~U[0.9, 1.2], and τ_3~U[1.8, 2.2]. α_1, α_2, and α_3 obey a uniform marginal distribution in the range from 0 to 1 and are subject to $\alpha_1 + \alpha_2 + \alpha_3 = 1$. Fig. 3.7 (b) shows the training and validation MAE for the five parameters. All of them are well converged at MSEs below 0.01 after 200 epochs. The testing results of new data with the same ranges as the training data are shown in Fig. 3.7 (c). Astonishingly, the v has high prediction precision for all parameters with SNR ranging from 26 ~ 44 dB. In contrast, in Fig.3.7 (d), LSF delivers poor results for all parameters, especially the α_1 and α_2. The performance of LSF nearly keeps unchanged within the whole SNR range.

Table 3.2. Fitting Performance Comparison

bi-exponential decay model					
Parameter	τ_1	τ_2	α		
F (CNN)	**35.5**	**5.0**	**79.8**		
F (LSF)	82.6	19.7	640		
Precision Gain	**5.3**	**15.2**	**25.0**		
tri-exponential decay model					
Parameter	τ_1	τ_2	τ_3	α_1	α_2
F (CNN)	**35.1**	**7.5**	**3.7**	**905.5**	**1.75e3**
F (LSF)	403	44.3	36.3	6.7e3	4.2e3
Precision Gain	**132.3**	**34.8**	**96.0**	**54.8**	**5.8**
average speed (ms/sample)					
CNN	0.12				
LSF	37.5				
Speed Enhancement	**≈ 300**				

We further quantitatively investigate the gains of the CNN algorithm against LSF in terms of precision and computational speed. To systematically compare CNN and LSF, here we adopt the figure-of-merit (*F*-value) as an indicator. *F*-value is defined as [143]:

$$F = \frac{\delta x}{x} SNR, \tag{3.5}$$

where x is a lifetime parameter (τ_{i} or α_i) and δx is its standard deviation. F is a normalized value for assessing the sensitivity of a lifetime parameter determination algorithm. An ideal F = 1, and usually F > 1. Hence, the precision gain can be simply defined as $G_x = [F_x^{LSF}/F_x^{CNN}]^2$. Table 3.2 compares the fitting/prediction performances and calculation speeds of CNN and LSF for bi- and tri-exponential decay models. The F-values were obtained from the testing datasets for bi- and tri-exponential decay models with 40,000 samples in Figs. 3.4 (d) and 3.7(c). It shows that our CNN significantly outperforms traditional LSF with compelling precision gains. Besides, CNN shows a 300-fold enhancement in the calculation speed under the same hardware platform (Intel i7-4790 CPU).

Unlike 2D or 3D CNNs that usually require high-performance GPUs with a massive number of parallel processors or cloud computing, training the proposed 1D-ConvResNet for bi-, tri-, or even multi-exponential decay models can be easily performed with desktop computers. This is due to the highly efficient neural network architecture. Table 3.3 compares the proposed CNN with previously reported high-dimensional CNNs and feed-forward dense neural networks. The total parameters are only about 4.5% of the 3D FLINET, indicating that large memory and floating-point operations can be saved. The training procedure took only tens of minutes on a laptop with the Intel i7-4790 CPU. In contrast, higher-dimensional CNN architectures usually take several hours, even with GPU acceleration. Much shorter training time is beneficial because it ensures that CNN can be rapidly re-trained and deployed for different FLIM applications.

Table 3.3. Architecture Comparison

	1D-CNN	FLINET[3]	DENSE NET[2]
Total Parameters[a]	**48,675**	1,084,045	149,252
Resolving multi-exponential decay models?	**Yes**	Yes	No
Training platform	**CPU**	GPU	CPU

Training time^b (h)	**0.5**	4	4

[a] For a fair comparison, the time bin is set to 256 for all three networks. The neuron number of the 4-layer DENSE NET is set as 256, 256, 256, and 4 since the authors did not show exact neuron numbers in [130].

[b] The training processor is Intel i7-4790 CPU for our network, NVIDIA TITAN Xp GPU for [132], and Intel(R) Xeon(R) E31245 for [130].

3.5 Evaluation of experimental FLIM data

The proposed 1D CNN was further employed to analyze two-photon FLIM images of functionalized gold nanoprobes in living cells. GNRs support shape-dependent localized surface plasmon resonance, leading to tunable optical properties and strong two-photon luminescence [144]. GNRs are efficient and stable fluorescence quenchers for organic fluorophores [145]. Gold nanoprobes are GNRs functionalized by fluorophore-labeled hairpin oligonucleotides, which have found various biomedical applications including RNA detection for cancer diagnosis, research of energy transfer behaviors, and recognition of analytes [146, 147]. This study's GNRs were functionalized by Cy5-labeled thiolated single-strand DNAs (Cy5-ssDNA-GNRs) via ligand exchange and salt aging processes. To identify the locations of endocytic gold nanoprobes in living cells, FLIM images of Cy5-ssDNA-GNRs labeled Hek293 cells and human prostatic cancer cells were evaluated. Dr. Yu Chen’s research group provides the gold nanoprobes and cell samples at the department of physics, Strathclyde University. The gold nanoprobe synthesis and cell preparation process have been reported in [148].

3.5.1 FLIM data acquisition

Briefly, the cells were treated with gold nanoprobes solution for incubation. Then the samples were washed with phosphate-buffered saline (PBS) and fixed with paraformaldehyde. The sample was dispersed on a glass slide after staining with 4’-6-diamidino-2-phenylindole (DAPI), and then the coverslips were mounted for microscopy with a ProLong antifade solution. A two-photon FLIM system acquired FLIM images. It includes a confocal microscope (LSM 510, Carl Zeiss, Oberkochen, Germany), a femtosecond Ti: sapphire laser (Chameleon, Coherent, Santa Clara, USA), and a TCSPC acquisition system (SPC-830, Becker & Hickl GmbH, Berlin, Germany). The laser excitation wavelength is 850 nm, and the laser pulse width is less than 200 fs with an 80MHz repetition rate. The bin width of the TCSPC is 0.039ns, and each measured histogram contains 256-time bins. The emission light was collected by a 60× water-immersion objective lens (N.A =1.0) with a 500-550 nm bandpass filter.

3.5.2 Result Analysis

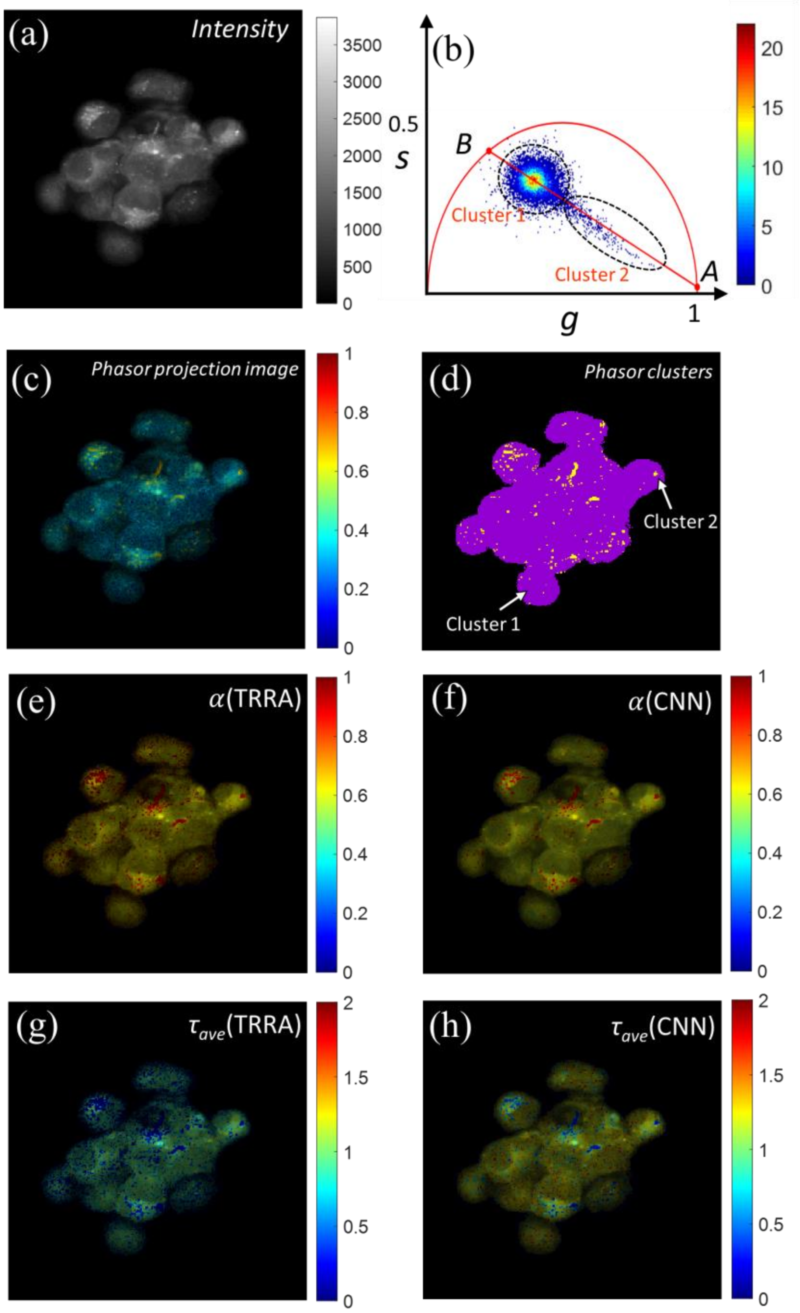


Fig. 3.8. FLIM images of Cy5-ssDNA-GNRs labeled Hek293 cells. (a) Intensity image; (b) phasor plot, the red line is the fitted line of the phasor pixels. A threshold with 20% of the maximum photon counts is set for fitting the line. *A* and *B* are the intersection points between the fitted line and the semicircle. Two dashed ellipses indicate two clusters; (c)-(d) the phasor projection cluster maps; (e)-(f) α maps and (g)-(h) τ_{ave} maps calculated by LSF and CNN.

The first case is the FLIM image of Cy5-ssDNA-GNRs labeled Hek293 cells. Fig. 3.8 (a) shows the grey-scale fluorescence intensity image. Gold nanoprobes show higher intensities than cell tissues due to their strong two-photon luminescence. However, it is difficult to precisely identify their locations solely from intensity images because the amount of gold nanoprobes is quite small, and the fluorescence intensities of cells have a large variance. The two-photon luminescence of GNR has ultra-short lifetimes of around tens of picoseconds [149]. The free state Cy5 has a lifetime of around 2 ns. When Cy5 is bounded in a closed hairpin oligonucleotide structure attached to a gold surface, Cy5 is in a quenched state. It has a short lifetime in the sub-nanosecond range because of the intense energy transfer between Cy5 and GNRs. Hence, gold nanoprobes show complex ultra-short lifetimes. Fig. 3.8 (b) shows the corresponding phasor plot of the FLIM image. A threshold of 100 is set to filter out low-intensity pixels. Two clusters can be observed from the phasor plot. Cluster 1 with the most pixels is the major cluster located near B, showing that these pixels have longer lifetimes. Another cluster with a tail extending to the right bottom can also be found. The number of pixels in Cluster 2 is much smaller, and the lifetimes are shorter. The whole phasor pixels nearly show a straight-line distribution, suggesting that a bi-exponential decay model can analyze the FLIM image. Before conducting CNN or LSF analysis, the phasor orthogonal projection and cluster images of the FLIM data were calculated (see details in Appendix). Since there is no ground truth for experimental data, the phasor and cluster images can be a reference for comparing the LSF and CNN results. Fig. 3.8 (c) shows the pseudo-color image of the phasor projection score β, in which each phasor pixel is projected on the line segment between A and B with β being the relative distance from point B. The two points A and B are the intersection points between the fitted line for the pixels and semicircle. β changes from 0 to 1 as the projected position shifts from B to A. When the projected position is out of the line segment, β is set to 0. It should be noted here that β has a different meaning to α because the pixels in the phasor plot follow a nonlinear spatial distribution from the Fourier transform. Nevertheless, β is sufficient to distinguish gold nanoprobes from cell tissues in Fig. 3.8 (c). Meanwhile, β can also be used as a criterion to segment Clusters 1 and 2. Fig. 3.8 (d) shows the corresponding cluster image for β = 0.4; the pixels in Cluster 1 are attributed to the autofluorescence of the cells whereas the pixels in Cluster 2 are attributed to the endocytic gold nanoprobes. Figs 3.8 (e)-(f) and (g)-(h) show the calculated α maps and amplitude-weighted average lifetime (τ_{ave}) maps using LSF and CNN, respectively. The intensity threshold is 100 photon counts. Since the pixels of nanoprobes should have a larger α value, the red pixels in Figs 3.8 (e) and (f) indicate the locations of gold nanoprobes. Compared with the results in Fig. 3.8 (d), both LSF and CNN can obtain similar conclusions about the locations of gold nanoprobes in the phasor cluster analysis. However, the α map calculated by CNN can more accurately identify gold nanoprobes, whereas LSF-estimated images are noisier due to a larger deviation range. It becomes worse for LSF to resolve τ_{ave}. Fig. 3.8 (g) shows the τ_{ave} map calculated by LSF; the blue pixels with shorter lifetimes indicate the locations of gold nanoprobes. It is difficult to identify gold nanoprobes from Fig. 3.8 (g), suggesting that LSF performs worse in resolving τ_{ave}. In contrast, CNN-estimated τ_{ave} and α maps agree with the phasor cluster analysis, more accurately locating gold nanoprobes.

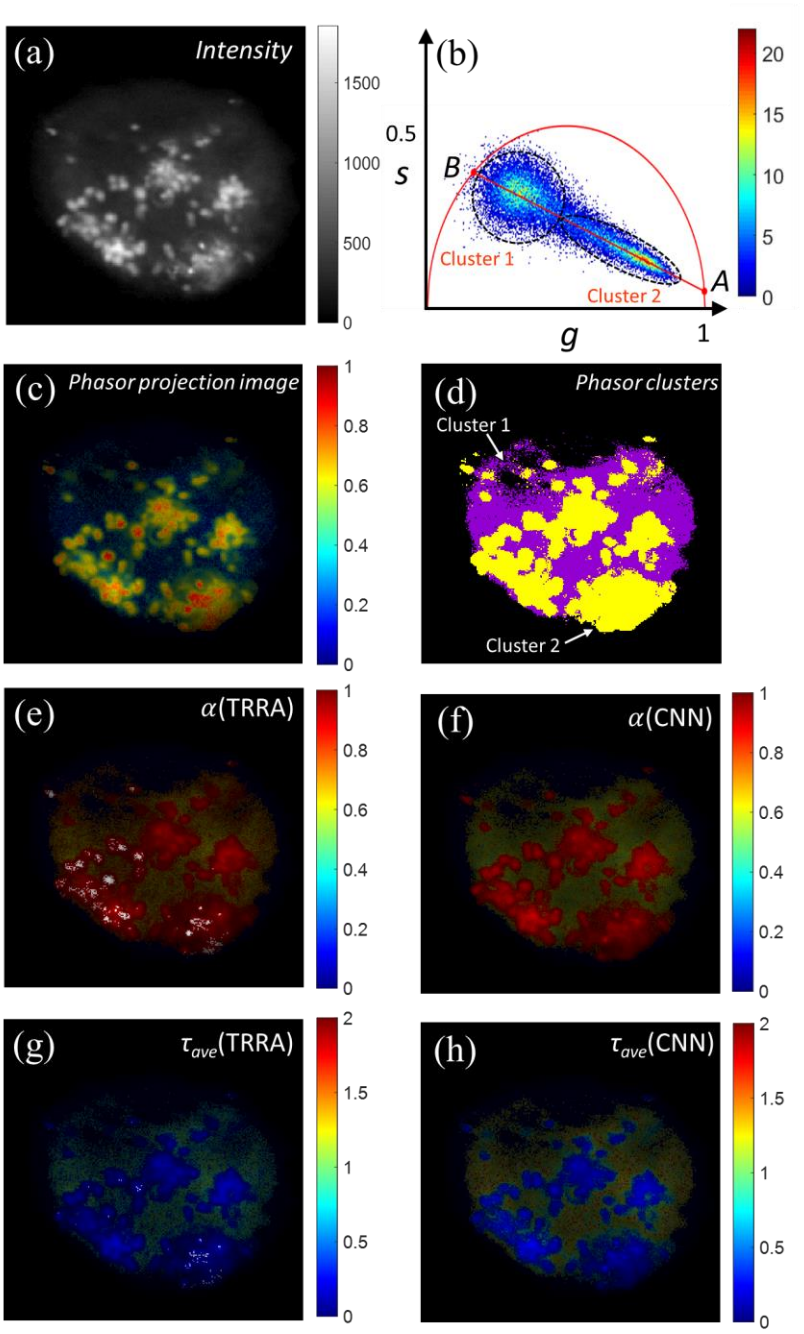


Fig. 3.9. FLIM images of a single human prostatic cancer cell with endocytic Cy5-ssDNA-GNRs. (a) Intensity image; (b) phasor plot. (c)-(d) phasor projection and cluster maps; white pixels mean that α is beyond [0, 1]; (e)-(f) α maps and (g)-(h) τ_{ave} maps obtained by LSF and CNN.

The second case shown in Fig. 3.9 is the FLIM image of a single human prostatic cancer cell with endocytic Cy5-ssDNA-GNRs. Different from Hek293 cells in Fig. 3.8, the average intensity is much lower and there are abundant endocytic gold nanoprobes. The intensities of gold nanoprobes are generally higher than 1000 similar to the first case. In contrast, only a few hundred average photon counts were collected from the cell in the second case. Fig. 3.9 (b) shows the corresponding phasor plot. The threshold is set to 100 counts. The pixel number in Cluster 2 shows a significant increase, attributed to an increased number of endocytic gold nanoprobes. Besides, the pixel distribution spreads to a broader range in Cluster 1 due to a low SNR level. Again, Fig. 3.9 (c) and (d) show the phasor projection and cluster maps, respectively. β is set to 0.4 to segment the two clusters. The LSF- and CNN-estimated α maps are shown in Fig. 3.9 (e) and (f), respectively. However, LSF fails to find converged estimations in many pixels (white pixel areas) in Fig. 3.9 (e). Such pixels correspond to the area with high concentrated GNRs and have ultra-short decays, leading to divergent solutions in the deconvolution procedure. The LSF -estimated τ_{ave} map shown in Fig. 3.9 (g) also fails to identify gold nanoprobes robustly. Instead, CNN provides a clearer τ_{ave} map in Fig. 3.9 (h), more accurately identifying gold nanoprobes. The autofluorescence background in the cell was also resolved compared with Fig. 3.9 (g). It is shown that CNN is more robust under low-count conditions.

3.6 Conclusion

In this chapter, we propose the 1D-CNN algorithm for FLIM data analysis. The design, training, and quantitative evaluation are presented in detail. The 1D-CNN algorithm has unparallel advantages including less architectural complexity, hardware-friendly, fast training, and fast calculation speed. Our algorithm can be easily trained using synthetic FLIM data for different lifetime ranges. The study shows the feasibility of our proposed CNN for fast calculating the FLIM image with high accuracy. The analysis of simulated FLIM data shows 1D-CNN algorithm has superior performance on FLIM images with a large range of intensity levels. The trained algorithm is then applied to analyze the experimental data of two-photon FLIM images of functionalized gold nanoprobes in hek293 and human prostate cancer cells. The results confirm the outstanding performance and robustness of our proposed algorithm. In a later chapter, the 1D-CNN algorithm will be further optimized using quantized neural network architecture and implemented on FPGA to accelerate the speed for fast FLIM analysis.

Chapter 4

Deep Learning for Improving Spatial Resolution of Fluorescence Lifetime Imaging

4.1 Introduction

This chapter presents the development of the DL algorithm to obtain HR fluorescence lifetime images from LR images acquired from FLIM systems. Chapter 3 demonstrated that the pixel-wise DL algorithm can fast and accurately estimate fluorescence lifetimes. This chapter further extends the DL's functionality to improve the spatial resolution of FLIM, thus directly obtaining HR images from LR 3D FLIM data. It is a great challenge to acquire HR images rapidly in current FLIM systems. Chapter 2 introduces two different instrumental configurations for obtaining FLIM images: laser scanning microscopy (LSM) and wide-field illumination microscopy (WFIM) [51]. LSM FLIM systems acquire images by scanning the focal point across samples. Obtaining HR images requires a minimized focal point and a smaller scanning step, leading to a long acquisition, photobleaching fluorophores, and causing phototoxicity. On the other side, WFIM systems adopt a parallel light field to excite samples, offering a higher frame rate and causing less damage to samples. However, WFLIM’s spatial resolution is poor, and images are easily contaminated by out-of-focus surrounding light. Besides, there is a trade-off between the spatial and temporal resolutions due to the crosstalk [150].

New sensor and microscopy technologies have been introduced to improve FLIM systems’ performances, including short-deadtime high-throughput single-photon sensors and multifocal scanning systems [62, 151]. This chapter proposed an alternative computational strategy using DL techniques to rapidly obtain HR FLIM images from raw image data without modifying the system configuration. Our method combines DL-based fluorescence lifetime determination and single image resolution enhancement (DL-SIRE) techniques. DL techniques provide a new approach to delivering fast analysis with different models [132, 152, 153]. They have achieved unprecedented accuracy and fast speed. 'SIRE' refers to reconstructing a single HR image directly from its LR counterpart. It is a notoriously ill-posed problem because an infinite number of HR images correspond to a given LR image. It should be noted that 'SIRE' often refers to 'Single Image Super resolution' (SISR) in the computer vision community [154, 155]. However, super-resolution usually means achieving spatial resolution beyond the diffraction limit in biomedical and microscopy communities. We used 'SIRE' to avoid confusion instead of 'SISR'. DL's powerful hierarchical representation learning ability allows DL-SIRE to reconstruct state-of-the-art HR images by learning a single image's intrinsic and high-level statistical features. DL-SIRE has been widely used in face recognition, remote sensing, and video surveillance [156, 157]. It has become a research trend in improving the spatial resolution of optical and fluorescence microscopes [113, 114, 158-160]. However, to our knowledge, DL-SIRE has not yet been used in FLIM systems, and there are two main challenges. First, there is a lack of massive FLIM data for training neural networks. Unlike natural images, FLIM images are three-dimensional (spatial and temporal) data. Acquiring a large number of FLIM images is labor-intensive and time-consuming. Second, acquiring high-resolution FLIM imaging takes much longer as it requires collecting enough photons and maintaining the resolution.

The main contributions to addressing the above challenges include:

1. We proposed a method to generate a large-scale semi-synthetic FLIM dataset for training neural networks. The samples in the FLIM dataset include distinct morphological characteristics of subcellular structures and organelles and complex decays with a wide range of fluorescence lifetimes.

2. A hybrid neural network architecture, SRI-FLIMnet, was proposed to map 3D LR FLIM data to corresponding HR images. SRI-FLIMnet serves two purposes: A) to directly map a decay histogram to its corresponding lifetime without iterative optimization processes and B) to nonlinearly transform LR FLIM to HR FLIM images. LR images can be fast obtained through both LSM and WFIM FLIM systems. SRI-FLIMnet was evaluated using semi-synthetic FLIM images and validated with experimental FLIM data. Results confirm SRI-FLIMnet's superiority in reconstructing HR images. SRI-FLIMnet is expected to fast capture HR FLIM images and find potential applications in high-throughput and real-time FLIM systems.

4.2 Semi-synthetic training dataset preparation

Deep learning techniques are data-driven approaches. A large amount of suitable training data is necessary for developing a robust DL algorithm. However, for DL tasks in microscopy, there are limited datasets. Unlike open-source natural image datasets used for DL computer vision applications such as CIFAR-10, MS COCO, and DIV2K [161], most previously reported studies only considered specific tissues or cells, consisting of dozens to hundreds of samples for network training [113, 114, 158-160]. It should apply data argumentation techniques to increase the training data to train the network [162]. In addition, it is challenging to generalize previously reported DL algorithms for general applications. The lack of training datasets severely handicaps the development of DL FLIM techniques. Considering that FLIM data possess many spatial and temporal features, traditional data argumentation techniques cannot achieve desirable performance. One effective method to circumvent this difficulty is to generate model-based synthetic data to train networks. Data synthesis is a mature and widely used method to approximate realistic data and serve network training purposes. It has been successfully applied to many DL tasks, including light detection and ranging (LiDAR), fluorescence lifetime estimation, and super-resolution microscopy [132, 152, 159]. This section proposed a theoretical framework to obtain massive semi-synthetic FLIM data. The forward TCSPC model was first developed to describe the time-stamped photon detection process. Then semi-synthetic FLIM images were generated by converting immunofluorescence intensity images from the Human Protein Atlas (HPA) dataset to 3D FLIM data using the TCSPC model. In addition, the degrading model between HR and LR images was also developed to obtain LR-HR training pairs. We will demonstrate that semi-synthetic data can be used for neural network training and successfully applied to different experimental data.

4.2.1· TCSPC model

In TCSPC, a train of laser pulses excites the sample periodically. In each cycle, at most one photon is detected by a single-photon detector to avoid the pile-up effect. According to the photon counting theory, a non-homogeneous Poisson process govern such a process[163]. For a pixel (i, j) in an *M*×*M*×*L* FLIM image, the rate function $\lambda_{i,j}$ can be expressed as:

$$\lambda_{i,j} = A_{i,j} \cdot I\mathrm{RF}(t) * \sum_{p=1}^{D} \alpha_p \, e^{-t/\tau_p} + \epsilon, \quad i, j \in [1, M], t \in [1, L], \tag{4.1}$$

where *M* is the length of spatial dimension and *L* is the number of recorded time bins. $A_{i,j}$ is the detected fluorescence amplitude at pixel (i, j), which relates to the emitted fluorescence intensity and the detector's quantum efficiency; *IRF*(*t*) is the instrument response function; α_p is the fraction ratio of the lifetime component τ_p, and *D* is the number of lifetime components. The asterisk * denotes the convolution operator. An additional term ϵ represents added noise from surrounding light and the sensor's dark count noise. We assume fluorescence signals are collected at the focal plane for simplicity without considering the spatial blur function. After *S* cycles, the detected photon number at pixel (*i, j*) follows a Poisson distribution:

$$N_{i,j} \sim Poisson\left(S \int_0^T \lambda_{i,j}(t)dt\right). \tag{4.2}$$

The *N* photon arriving times denoted by $t_{i,j}^{n}\left(n \in \left[1, N_{i,j}\right]\right)$ at pixel (i, j) are independent and identically distributed random variables with the probability-density function described as:

$$f_{i,j}(t) = \frac{\lambda_{i,j}(t)}{\int_0^T \lambda_{i,j}(t)dt}. \tag{4.3}$$

Eqs. (4.1) - (4.3) define the TCSPC forward model. Given the decay model and lifetime parameters, we can simulate the recorded timestamps of the first *N* arrival photons using the acceptance-rejection method. As low noise sensors have been used, Poisson noise is dominant in modern TCSPC measurements. We have compared synthetic and measured decays (analysis not included), showing that the TCSPC model can precisely describe fluorescence decays with arbitrary SNR levels.

It should be noted that in Chapter 3, we use the additive noise model and Monte Carlo simulation to generate training datasets. This method can generate massive FLIM data fast and efficiently. However, the additive noise model is an approximate method, which is only suitable for high photon conditions. Due to the discrete nature of photons and the random feature of noise, this method cannot simulate a decay histogram with given photon counts. In contrast, the TCSPC model is more precise. It can simulate fluorescence decay with any given photon count. But it also has disadvantages. The TCSPC model simulates every photon's statistical behavior, making it more computation-intensive and time-consuming.

4.2.2 Semi-synthetic FLIM images

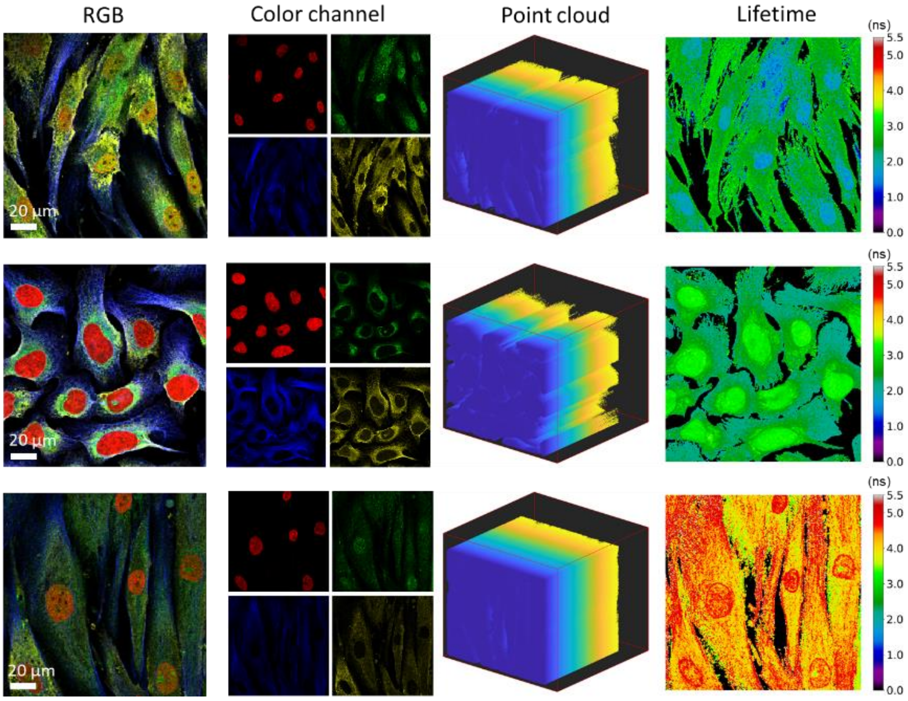


Fig. 4.1. Original intensity images in the HPA dataset and the semi-synthetic FLIM data. The 1st column is the RGB intensity images of three samples. Their corresponding four color-channel images are shown in the 2nd column; each organelle is labeled by one color. The 3rd and 4th columns show the synthetic 3D FLIM data cloud and lifetime images. In 3D point plots (3rd column), the dark blue to bright yellow point color indicates a high to low photon density.

The HPA dataset provides a comprehensive subcellular map of human proteomes in cells, tissues, and organs [164]. It contains more than 80,000 cellular images with a large morphological diversity obtained by confocal immunofluorescence microscopy systems. The images in HPA dataset cover a wide range of cellular morphologies. We used the HPA image classification dataset, including more than 30,000 samples with 27 different cell types [165]. Each sample contains four color channels, and each color channel contains one cellular organelle. The blue, red, yellow, and green channels contain nuclei, microtubules, endoplasmic reticulums, and proteins (such as endosomes, lysosomes, or mitochondria). Fig. 4.1 shows some original image samples in the HPA dataset and generated 3D FLIM data. The 1st column shows RGB images of three different samples, and the 2nd column shows their corresponding color-channel images. We can see that the cellular images contain more statistical features, including more refined details, complex spatial variations, and more sophisticated correlations than open-access natural images (mostly used by computer vision communities for DL training). Currently, there are no open-access FLIM datasets for developing DL-enhanced analysis tools. To generate FLIM data with complex decay features and various cellular morphologies, The samples in HPA datasets are converted to semi-synthetic FLIM data. The pixel values of intensity images are converted to time-stamped photon counts using the forward TCSPC model in Eqs. (4.1)-(4.3). The pixel values from 0 to 255 were linearly scaled from 0 to 5100. The photon count range corresponds to the typical intensity range in practical experiments and offers a sufficient SNR for accurate lifetime estimations. Instead of restricting to specified fluorophore lifetimes, each color channel was randomly assigned a single lifetime from 0.5 to 5.5 ns to include more general cases. The semi-synthetic FLIM data were obtained by combining all the considered color channels. Since different color channels have diverse intensities and spatial overlaps, each pixel naturally shows complex decays with different lifetime components in semi-synthetic FLIM images. Meanwhile, they also have a variety of shapes. The total photon count in a pixel of a FLIM image remains the same as the intensity image, while the temporal photon distribution follows an exponential decay with given lifetime parameters. In simulations, $M = L = 256$, $D = 1$, and the total recorded time is 10 ns. The IRF is modeled by a Gaussian distribution with a Full Width at Half Maximum (FWHM) = 167.3 ps. It matches the two-photon FLIM system we used for experiments (see details in Section 4.5). The added noise can be reasonably neglected because Poisson noise dominates in TCSPC measurements. Additionally, in FLIM analysis, a lower threshold is usually set to exclude pixels with low photon counts. The 3rd and 4th columns of Fig. 4.1 show the generated data cloud and the corresponding fluorescence lifetime images calculated by the least-square fitting method, respectively. Each point corresponds to a detected photon's spatial and temporal positions in the data cloud. FLIM images have lower contrast than RGB-intensity images in revealing cell structures. Many delicate structures are blurred or even disappear, and this is because fluorescence lifetime is independent of the intensity variation. FLIM data with only one, two, or three decay components are also generated to ensure diversity. The dataset contains 30,000 FLIM samples for training purposes, providing sufficient data for network training.

4.2.3 Degrading model

An accurate degrading model for correctly developing LR-HR image pairs is crucial for DL-SIRE's performance. For general DL-SIRE frameworks, the degrading model is[155]:

$$I^{LR} = D_k(I^{HR} * K) + \varepsilon, \tag{4.4}$$

where I^{LR} and I^{HR} denote the LR and HR images, respectively, K denotes the blurring kernel, and D_k is the down-sampling operator with a scaling ratio of k. ε is added noise. The bicubic interpolation is usually applied in typical computer vision tasks as the degrading model [154]. The Gaussian blurring kernel is widely used in super-resolution microscopy tasks [113, 158]. However, the situation is different in FLIM systems. In LSM FLIM systems, LR images are measured using coarser scanning steps. Therefore, the LR image, I^{LR} with the scaling ratio k, is obtained by summing a patch of k×k pixels in the corresponding HR image, I^{HR}. It also applies to WFIM-FLIM systems, where a low-resolution wide-field camera like a SPAD array is used[166]. LR images are equivalent to spatially pixel-binned HR images. Thus, the degrading operation can be described as a three-dimensional convolutional layer, and the degrading model is expressed as:

$$I^{LR}(m, m, L) = 1^{k\times k\times 1} \otimes_{s=(k,k,1)} I^{HR}(M, M, L), \tag{4.5}$$

where $1^{k\times k\times 1}$ denotes k×k×1 matrix with all one elements, $\otimes_{s=(k,k,1)}$ is the three-dimensional cross-correlation operator with the stride s = $(k, k, 1)$ along the spatial and temporal dimensions. The size of I^{LR} is m×m×L with m = mod (M, k). For simplification, the intensity changes in LR pixels are not considered because the fluorescence lifetime is independent of the intensity, given the SNR is large enough. Eq. (4.5) reveals that in LR images, low- and high-frequency information is lost through spatial pixel binning. Meanwhile, the temporal information among neighboring pixels fuses, resulting in information loss and a shift in the lifetime distribution. Achieving HR FLIM is more challenging compared to other situations. In semi-synthetic FLIM data processing, when applying Eq. (4.5) to HR images, a threshold of 100 photon counts was set for each color channel image to exclude low-intensity pixels.

4.3 Network design and training

4.3.1 Network architecture

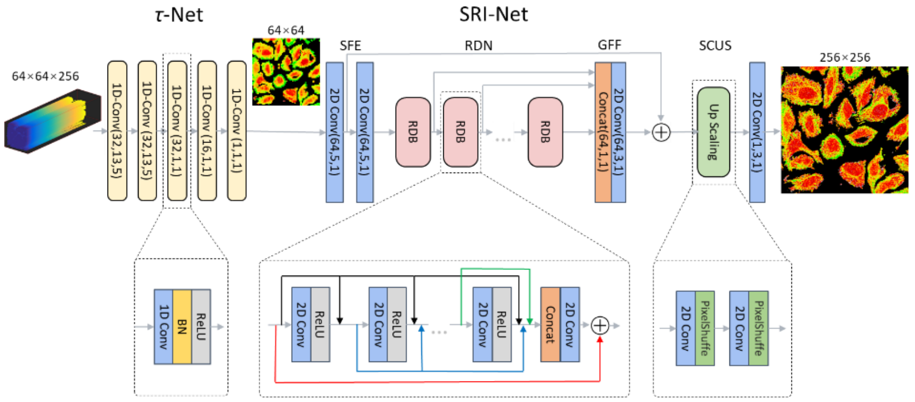


Fig. 4.2. Network architecture of the proposed SRI-FLIMnet with a scaling ratio $k = 4$. SRI-FLIMnet contains two subnets, τ-Net, and SRI-Net, responsible for lifetime evaluation and nonlinear transformation for improving spatial resolution. The parameters of convolutional layers are labeled in the parentheses as (feature number, kernel size, and stride). SRI-Net contains eight residual dense blocks (RDB) with hybrid dilated convolution. In each RDB, eight 2D convolutional layers with a kernel size of 3 and a stride of 1 are densely connected. Both the input feature number and growth rate are 64. In the Up-Scaling part of SRI-Net, the parameters are (1024, 3, 1) for both convolutional layers. The input data of SRI-FLIMnet is 64×64×256, and the final output data size is 256×256.

It has been a great challenge to simultaneously process the spatial and temporal features of FLIM data. One intuitionistic approach is to design a three-dimensional neural network architecture to map 3D FLIM data to HR images. However, the spatial and temporal information could interfere, leading to difficulty in network training. It would also cause an enormous computation burden and a prolonged inference speed. Our proposed SRI-FLIMnet adopts a temporal and spatial decoupling approach to address this problem. The SRI-FLIMnet, as shown in Fig. 4.2, uses a hybrid network architecture composed of two sub-nets: τ-Net and SRI-Net. The former aims to estimate fluorescence lifetimes, whereas the latter is to produce spatial-resolution improved lifetime images so that the crosstalk between spatial and temporal dimensions can be eliminated. τ-Net uses a simplified 1DCNN neural network to calculate pixel-wise fluorescence lifetimes. As demonstrated in Chapter 3, 1DCNN is efficient and hardware-friendly, suitable for fast analysis, and can accurately analyze single- and multi-exponential decay models. τ-Net contains five 1D convolutional blocks. Each block has a convolution layer, a batch normalization layer, and a nonlinear activation layer (ReLU). The hyperparameters of all layers in τ-Net are annotated in Fig. 4.2. We adopt the residual dense network (RDN) with hybrid dilated convolution as the backbone for SRI-Net[167]. RDN can fully exploit local and global hierarchical features from LR images. It consists of four parts, the shallow feature extraction (SFE) layers, residual dense blocks (RDB), global feature fusion (GFF) layers, and the sub-pixel convolution up-sampling (SCUS) block. SFE layers extract high-level features of input LR images for later network blocks. RDBs are for learning and fusing local features with different levels. In an RDB, eight 2D convolutional layers are densely connected. The input feature of the first layer is 64, and the growth rate of the feature number in the following layers is 64, too. A concatenation layer is put at the end of the RDB, followed by a residual connection across the whole block. To achieve a better performance, RDN requires a deeper architecture with many RDBs to obtain a large receptive field and high nonlinearity. The receptive field is a critical factor in the network's performance because it decides how much contextual information is used to reconstruct high-frequency components. However, a deeper architecture consumes more computation resources and is, therefore, slower in training and inference. To overcome this problem, we further improved the network by introducing the hybrid dilated convolution in each RDBs. The dilated convolution with dilation rate *l* is expressed as:

$$(F *_l K)(p) = \sum_{s+lt=\boldsymbol{p}} F(s)K(t), \tag{4.6}$$

where $F: Z^{\mathbb{2}} \to R$ is a discrete function, $K: \Omega_r \to R$ is the convolution kernel with $\Omega_r = [-r,r]^2 \cap Z^{\mathbb{2}}$. By inserting zeros in the convolution kernel, dilated convolutions can exponentially expand the receptive field without extra parameters. In each RDB, the dilation rate for 8 convolutional layers is [1, 2, 5, 9, 1, 2, 5, 9]. We particularly used mixed saw-toothed dilation rates for different layers to expand the convolution kernels to different scales while avoiding the gridding effect and information loss brought by dilated convolutions[168]. The GFF block is for global feature fusion and global residual learning, and the final SCUS block up-samples images to the size of corresponding HR images. Fig. 4.2 only shows the network architecture for a $k = 4$. Note that for $k = 2$ and 3, there are only one 2D convolutional layer and one Pixel-Shuffle layer in the Up-Scaling part for obtaining the desirable data size of output images. The corresponding feature number of the convolutional layer is $64\times k^2$, and the kernel size and stride remain 3 and 1, respectively.

4.3.2 Loss function

For τ-Net, the Mean-Square Error (MSE) loss or L_2 norm loss is used for network training:

$$L_2(\hat{\tau},\tau) = \frac{1}{N}\sum_{i=1}^{N}(\hat{\tau}_i - \tau_i)^2\,, \tag{4.7}$$

where $\hat{\tau}$ is the predicted lifetime by τ-Net and N is the batch size. As for SRI-Net, although the L_2 loss tends to obtain a high peak signal-to-noise ratio (PSNR), the reconstructed images are over-smoothed and unappealing to human visual perception. L_2 also inherently assumes that noise follows a Gaussian distribution. It is hard to conform to this condition in our study. Therefore, we adopted the perceptual loss to restore high-frequency contents as much as possible. The perceptual loss, hereafter named L_{percep}, is composed of two terms, the multiscale structural similarity index measure (MS-SSIM) loss and L_1 loss, which is expressed as[169]:

$$L_{percep}\left(\hat{I}^{HR}, I^{HR}\right) = L_{MS-SSIM}\left(\hat{I}^{HR}, I^{HR}\right) + \gamma L_1\left(\hat{I}^{HR}, I^{HR}\right), \tag{4.8}$$

where $\hat{I}^{HR}$ denotes the reconstructed HR image by the network. $L_{MS\text{-}SSIM}$ is defined as:

$$L_{\mathrm{MS-SSIM}} = 1 - \left[l_{\mathrm{Ms}}\left(\hat{I}^{HR}, I^{HR}\right)\right]^{\alpha_{Ms}} \prod_{j=1}^{Ms}\left[cs_j\left(\hat{I}^{HR}, I^{HR}\right)\right]^{\beta_j}, \tag{4.9}$$

where $l(\hat{I}^{HR}, I^{HR}) = (2\mu_{\hat{I}^{HR}}\mu_{I^{HR}} + C_1)/(\mu_{\hat{I}^{HR}}^2 + \mu_{I^{HR}}^2 + C_1)$ is the luminance comparison function with μ the mean of $\hat{I}^{HR}$ or I^{HR}. $cs(\hat{I}^{HR}, I^{HR}) = (2\sigma_{\hat{I}^{HR}}\sigma_{I^{HR}} + C_2)/(\sigma_{\hat{I}^{HR}}^2 + \sigma_{I^{HR}}^2 + C_2)$ is the contrast-structure comparison function and σ is the standard deviation of $\hat{I}^{HR}$ and I^{HR}. C_1 and C_2 are constants to ensure stability when the denominator is close to 0. M_s denotes the resolution level of SSIM. To estimate MS-SSIM, $C_1 = 0.01^2$ and $C_2 = 0.03^2$ M_s = 5 and $\alpha = \beta = [0.0448, 0.2856, 0.3001, 0.2363, 0.1333]$. An 11×11 Gaussian sliding window with σ = 1.5 is used to calculate *l* and *cs* for different scales in a pixel-by-pixel manner. The overall $L_{MS\text{-}SSIM}$ is the average value of the whole image [169]. The L_1 loss serves as a content loss to enforce the reconstructed HR images close to the ground truth HR images. γ = 0.2 is a parameter to adjust the weight of L_1 losses.

4.3.3 Network training

We implemented SRI-FLIMnet using the *Pytorch* library in Python 3.8 and performed network training and inference on a workstation containing an Intel Core i9-10900X CPU at 3.7 GHz and an Nvidia Quadro RTX5000 GPU card with 16G memory. The two subnets were trained independently. For τ-Net, the training dataset contains 50,000 decay histograms generated from Eqs (4.1) – (4.3). For each histogram, the total photon counts vary from 100 to 10,000, and the number of lifetime components ranges from 1 to 4, with a lifetime ranging from 0.5 to 5.5 ns. All random variables follow uniform distributions. The optimizer is Adam with a learning rate of 1e-4, and the batch size is 256 [Section 2.7.2]. During the training process, an early stop with 20 patient epochs was added to prevent over-fitting. The total training time is about 5 mins for τ-Net. For SRI-net, the semi-synthetic dataset mentioned above was used for training. The sizes of LR FLIM data are 128×128×256, 85×85×256, and 64×64×256 for *k* = 2, 3, and 4, respectively, which are obtained using Eq. (4.5). The optimizer is Adam. The initial learning rate is 1e-4, dropping 0.9 times every 20 epochs. To fully use GPU memory, the batch sizes for the k = 2, 3, and 4 are 6, 12, and 24, respectively. The whole training epoch for different scaling ratios is 200, and the training time is about 400 h, 160 h, and 50 h for *k* = 2, 3, and 4. Once the networks were well trained, the inference time for different scaling ratios typically took several seconds to map the 3D LR FLIM data to 256×256 HR images.

4.4 Characterization of the neural network

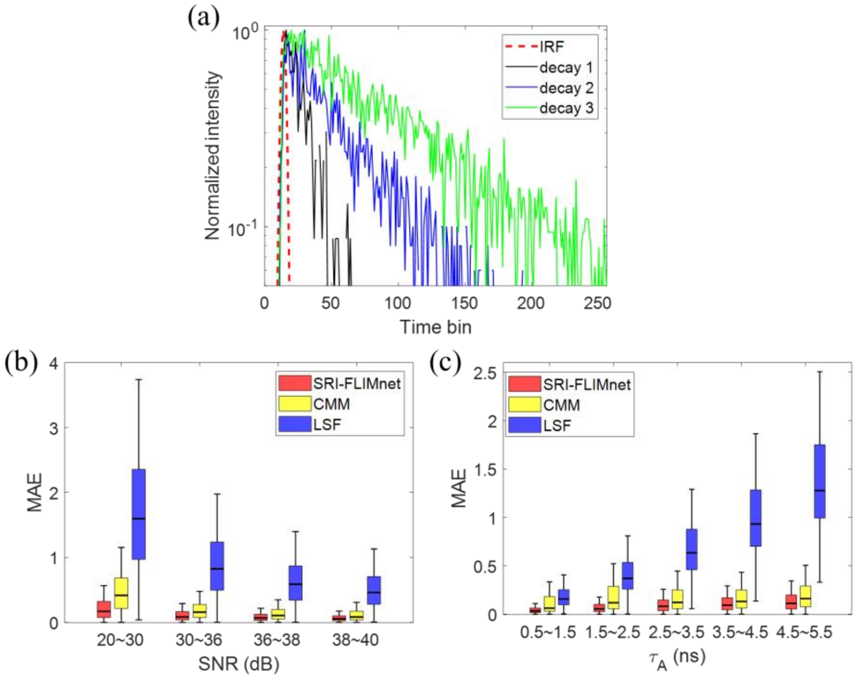


Fig. 4.3. Fluorescence lifetime determination performance of SRI-FLIMnet. (a) Decay samples used for testing τ-Net. The lifetime components for Decays 1 – 3 are 2, 3, and 4; the lifetimes for Decays 1 – 3 are [0.6, 1.5], [0.8, 2, 2.6], and [1.2, 1.8, 3.1, 4.5]; the total photon counts are 400, 2000, and 5000. The red dashed line represents the IRF. (b) Lifetime determination performance under different SNR conditions. Each group's lifetime ranges from 0.5 to 5.5 ns and the lifetime components randomly vary from 1 to 4. The SNR ranges from 20 ~ 40dB, corresponding to 100 ~ 10,000 photon counts. (c) Lifetime determination performance for different average lifetimes. In each group, the lifetime components randomly vary from 1 to 4, and SNR ranges from 20 ~ 40 dB. In (b) and (c), 2000 samples were considered in each group.

After the training procedure, SRI-FLIMnet was evaluated with a new testing dataset. The performance of SRI-FLIMnet on lifetime determination was first quantitatively assessed. Fig. 4.3 (a) shows three normalized decay histograms used for testing τ-Net. The samples denoted as Decays 1 – 3 have different SNR, lifetimes, and decay components. As mentioned above, the decay components may vary in different pixels in a FLIM image, and we do not know the exact number of decay components. Consequently, multi-exponential decay model analysis can cause model-mismatch problems and wrong interpretations. As suggested in [170], the amplitude-weighted average lifetime τ_A was used as a lifetime indicator, suitable for FLIM-

FRET measurements. τ_A is defined as $\tau_A = \sum_{i=1}^{D} \alpha_i \tau_i$, where D is the number of lifetime components. τ-Net directly predicts τ_A for input histograms. The accuracy of lifetime determination in terms of SNR is shown in Fig. 4.3 (b). The model-free algorithm, CMM and model-based LSF were also compared to SRI-FLIMnet. For LSF, the mono-exponential model is used for evaluating τ_A of multi-exponential decays. Because Poisson noise is the dominant noise source, the SNR of a decay histogram equals to $\sqrt{N}$ with N the total photon count. The SNR(dB) is calculated as $SNR(dB) = 20log\sqrt{N}$. The mean absolute error (MAE) determines the analysis accuracy. Fig. 4.3 (b) indicates that τ-Net shows the best accuracy, much better than traditional CMM and LSF. The median MAE of τ-Net is 0.169 for SNR 20 ~ 30 dB (100 ~ 1000 photon counts). As a comparison, the median MAE of LSF is 0.46 for SNR 38 ~ 40 dB (6400 ~ 10, 000 photon counts). The accuracy of τ-Net for low SNR is even better than that of LSF for high SNR. The lifetime determination performance over the whole dynamic lifetime range was investigated in Fig. 4.3 (c). Generally, given a fixed SNR, decays with larger lifetimes are nosier because photons are distributed in more time bins, resulting in fewer photons in a single time bin. Therefore, the accuracy is poor for decays with larger lifetimes. Nevertheless, τ-Net keeps a low MAE and small changes compared to the other two traditional methods over the lifetime ranges. The results in Fig. 4.3 indicate τ-Net is robust and accurate to calculate lifetimes. In addition, the calculation speed for τ-Net is more than 300-fold faster than LSF.

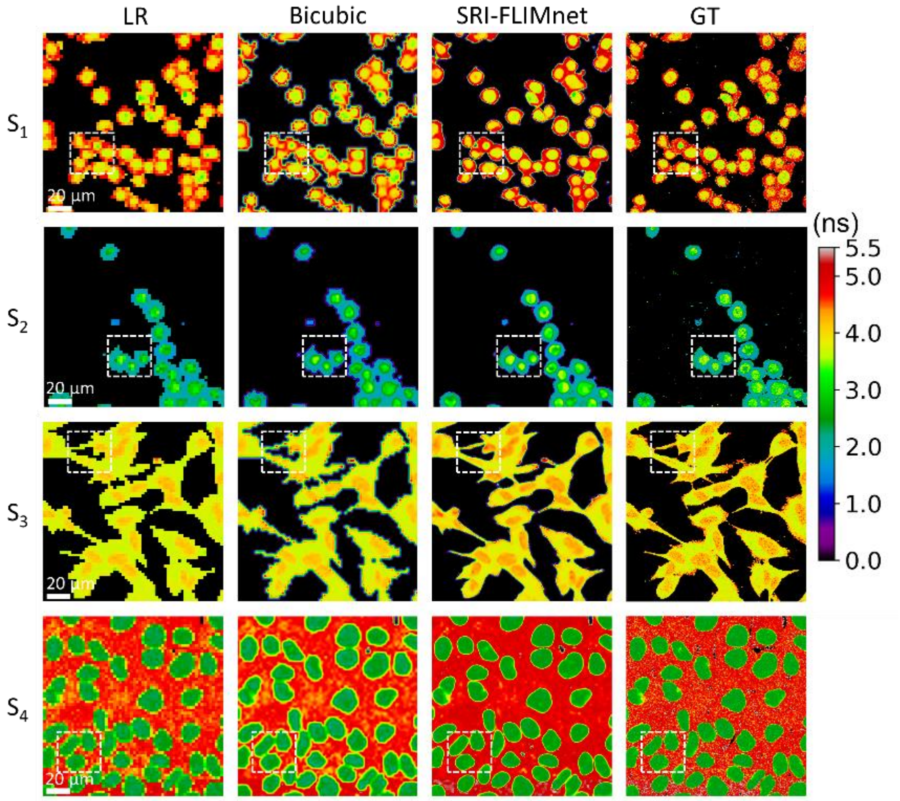


Fig. 4.4. Spatial resolution improved FL images of four samples with distinct morphology and lifetime distributions. The samples are denoted by S_1 – S_4. The low-resolution (LR), bicubic interpolation, reconstructed HR by SRI-FLIMnet, and ground-truth HR (GT) FL images are shown in the 1st row to 4th rows, respectively. The white dashed boxes in each subfigure depict the regions of interest (ROIs) for further analysis.

The performance of spatial resolution improvement for SRI-FLIMnet was further investigated. The testing FLIM dataset contains 1,000 samples. As an example, we evaluated reconstructed HR images with $k = 4$. In this case, suppose that the dwell time in each scanning step is fixed, and the image acquisition speed improves 16 times for LSM-FLIM systems. At the same time, LR images undergo severe degradation as much spatial and temporal information is lost. Fig. 4.4 shows the input LR, interpolated HR, SRI-FLIMnet-reconstructed HR, and ground truth (GT) FLIM images of four samples chosen from the testing dataset to illustrate the results better. The four samples labeled S_1 to S_4 include different morphological features such as a sizeable cellular population and complex cell patterns. They also have distinct lifetime distributions. As shown in Fig. 4.4, the 64×64 LR images predicted by τ-Net are shown in the 1st row, with a poor spatial resolution and a strong mosaic pattern. Compared with the GT images in the 4th row, the boundaries are unclear, and many delicate structures are missing. The lifetime distributions also have significant deviations. The 2nd row shows referenced interpolated 256×256 HR images using the bicubic interpolating method, indicating that the direct interpolation fails to improve the spatial resolution and introduces more artifacts. For example, the boundaries in all samples are severely distorted. In contrast, SRI-FLIMnet-reconstructed HR images in the 3rd row have sharper and smoother edges. They also recover more refined textures compared to LR images.

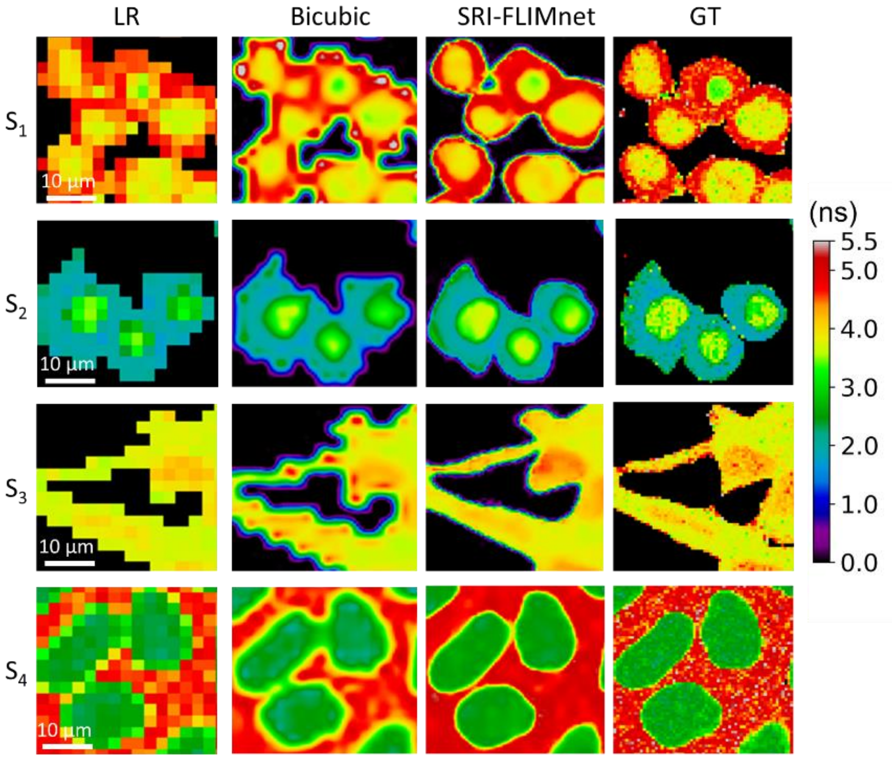


Fig. 4.5. Zoomed ROIs corresponding to the white dashed boxes in Fig. 3. The LR images lose most of the high- and low-frequency information and show strong mosaic patterns. The reconstructed HR images are more visually pleasing and close to the GT images than the interpolating method.

Fig. 4.5 shows the zoomed-in regions of interest (ROIs) marked by white dashed boxes in Fig. 4.4 to better illustrate the performance of SRI-FLIMnet. The results indicate that SRI-FLIMnet can generate shapes, boundaries, and general lifetime distributions closer to GT images from minimal information in LR images. SRI-FLIMnet-reconstructed images significantly outperform those using traditional interpolation methods. However, SRI-FLIMnet fails to reconstruct pixel-level refined structures, such as gaps between cells in S_1 and irregular boundaries in S_3. In addition, it cannot recover random lifetime variations in cells. For example, the spotty lifetime patterns in S_4 are blurred and smoothed. Therefore, SRI-FLIMnet has a solid ability to generate visually clear shapes and boundaries, whereas it cannot precisely reconstruct pixel-wise details and variations.

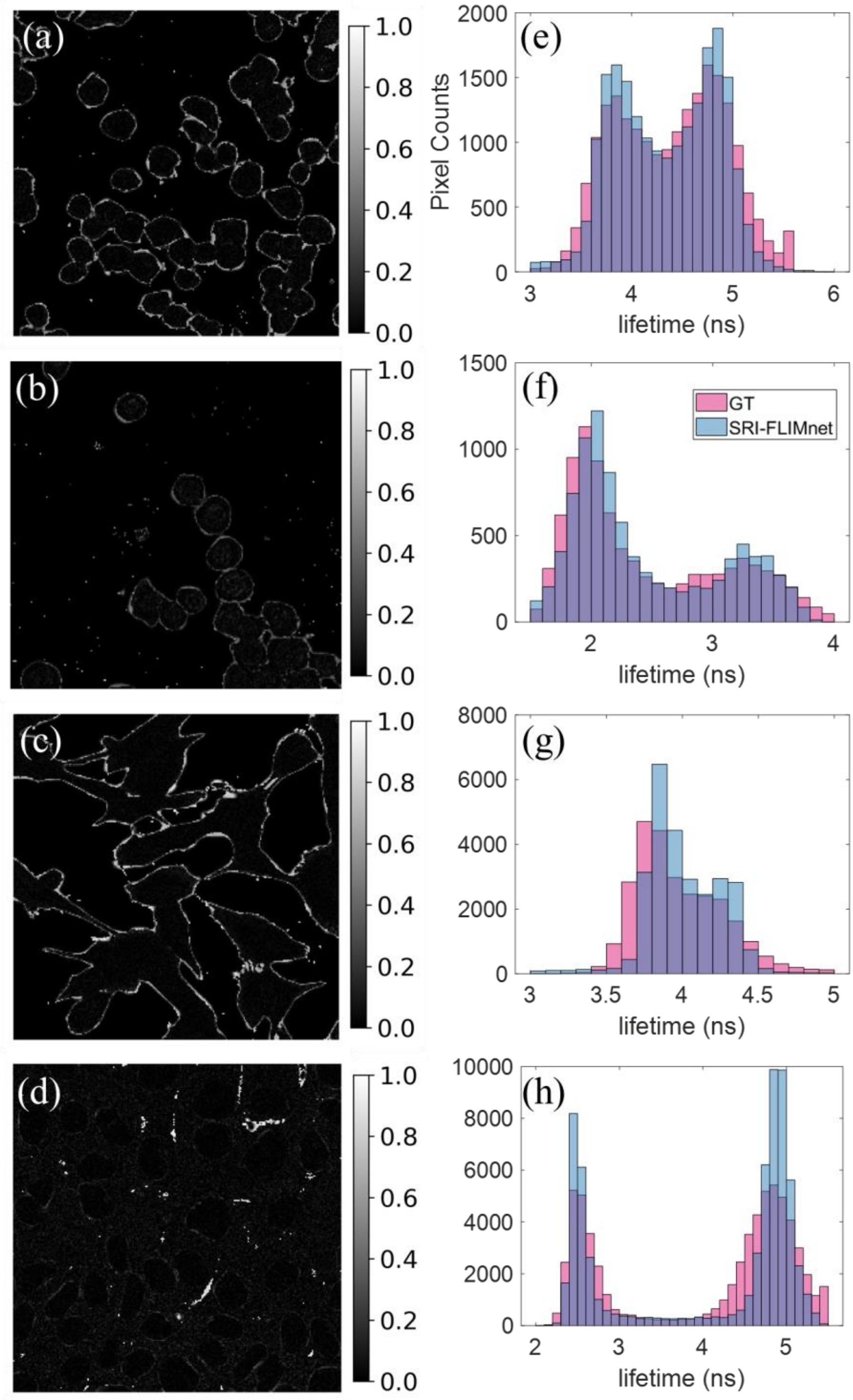


Fig. 4.6. Normalized absolute error maps (a) - (d) and lifetime distributions of the GT and SRI-FLIMnet reconstructed HR images (e) - (h) for $S_1 – S_4$. The normalized absolute error maps have an arbitrary unit.

The precision of SRI-FLIMnet-reconstructed HR images was also evaluated. Fig. 4.6 shows absolute error maps (Fig. 4.6(a) – (d)) and lifetime distribution histograms (Fig. 4.6 (e) – (h)) of $S_1 – S_4$ shown in Fig 4.4. The absolute error maps are the differences between reconstructed images and GT images. Since different samples have various lifetime ranges, absolute error maps are normalized to 0 ~ 1 for a better comparison. Interestingly, as shown in Fig. 4.6 (a) to (d), the largest errors always occur at outer boundaries for all four samples, where the spatial differentiation reaches its maximum value. The spatial differentiation has a significant effect on reconstruction precision. An area with sizeable spatial differentiation tends to have an apparent absolute error and vice versa. Figs 4.6 (e)- (h) show lifetime distribution histograms of $S_1 – S_4$, respectively. The two histograms of SRI-FLIMnet-reconstructed and GT images have primarily overlapped distributions for all four samples, revealing that SRI-FLIMnet can correctly restore the lifetime distribution in a wide dynamic range. However, there is a slight difference in lifetime distribution between reconstructed and GT images. The reconstructed images have more pixels with lifetime centering in the peak values. In addition, REC images' lifetime distributions are relatively narrower than corresponding GT images.

We further evaluated SRI-FLIMnet on the whole testing dataset. Table 4.1 summarizes the results of k = 2, 3, and 4. Two image quality metrics, SSIM and image MAE (IMMAE), were used for quantitative evaluation. The SSIM quantifies how SRI-FLIMnet can reconstruct close LR images to corresponding GT images. In contrast, the IMMAE measures the overall deviation of the lifetime distribution of reconstructed images from that of GT images. The IMMAE for an image is calculated as $|I^{Rec} - I^{GT}|/M^2$, where I^{Rec} and I^{GT} denote the reconstructed and GT images, respectively. M is the length of the spatial dimension. The dynamic lifetime range for calculating SSIM is 0 ~ 5.5 ns. In Table 4.1, the results are the average values of 10,000 samples. Both metrics of SRI-FLIMnet significantly outperform those of bicubic interpolation algorithms for different scaling ratios. In comparison, The SSIM of SRI-FLIMnet is approximately twice that of the bicubic interpolation method, whereas the

IMMAE of SRI-FLIMnet is also much better. The results reveal that SRI-FLIMnet performs better in reconstructing morphologies and lifetime distributions. It is worth noting that the testing samples we considered have complex structures and an extensive dynamic lifetime range. Both metrics for SRI-FLIMnet can be further improved if we only consider specified cell types and narrow the lifetime range.

Table 4.1. Quantitative Evaluation of Different Scaling Ratios on Testing Dataset

Scaling Ratio (k)	Bicubic interpolation		SRI-FLIMnet	
	SSIM	IMMAE	SSIM	IMMAE
2	0.33	1.41	0.73	0.20
3	0.32	1.43	0.64	0.26
4	0.31	1.44	0.58	0.31
GT	1	0	1	0

4.5 Experimental validations

4.5.1 Sample preparation

We validated the trained SRI-FLIMnet using FLIM data of bacteria-infected mouse raw macrophage cells cultured in DMEM (Dulbecco's modified Eagle medium) containing 10% FCS (fetal calf serum), maintained with 5% CO_2 in a humidified chamber at 37°C. Cells were seeded onto glass coverslips within 24-well cell culture plates. *Lactobacillus* and *Shigella Sonnei* bacteria were used in FLIM experiments. They were engineered to express a green fluorescent protein harvested at an early exponential phase. Cell cultures were infected at a multiplicity of infection of 100. After about one hour of infection, the wells were washed three times with PBS to remove extracellular bacteria. Then, cells were further incubated with DMEM containing 50 µs/mL gentamicin. The DMEM with gentamicin was removed at the chosen time, and cells were washed three times with antibiotic-free PBS. Cells were fixed with 3.7% paraformaldehyde for 15 minutes and permeabilized with 0.1% Triton X-100 for 5 minutes after being washed three times with PBS. Cells were stained with phalloidin Alexa Flour 546 after being washed three times with PBS and then were covered by coverslips with a ProLong antifade solution for microscopic examination.

4.5.2 Experimental data acquisition and analysis

A two-photon FLIM system was used to obtain FLIM data. The system contains a confocal microscope (LSM 510, Carl Zeiss), a femtosecond Ti: sapphire laser (Chameleon, Coherent) with an 850 nm wavelength as the excitation source, and a TCSPC acquisition system (SPC-830, Becker & Hickl GmbH) for time-tagging detected photons. The laser repetition rate is 80MHz, and the laser pulse duration is less than 200 fs. The fluorescence signals were collected using a 63× water-immersion objective lens (N.A. = 1.0) after passing a 500-550 nm bandpass filter. The TCSPC resolution is 0.039 ns, and each measured histogram contains 256-time bins. FLIM data (with a dimension of 256×256×256) were taken as ground truth HR images. As the scanning resolution cannot be adjusted in our FLIM system, we obtained LR FLIM data through pixel binning as described in Eq. (5). We used a $k = 4$ as an example. Fig. 4.7 shows the results of 4 different samples denoted by S_1 to S_4. The peak intensities of the samples range from 4,000 to 16,000 photon counts, providing a sufficient SNR for determining lifetimes. The bacteria have a smaller lifetime of around two ns, contrasting to macrophage cells. Therefore, FLIM images can better identify cells infected by bacteria and localize bacteria. The 1st column shows LR lifetime images of the samples. The 2nd and 3rd columns are reconstructed HR images by bicubic interpolation and SRI-FLIMnet, respectively. The last column is GT images for reference. SRI-FLIMnet can generate clear boundaries, whereas bicubic interpolated HR images show distinct jagged artifacts. This can be observed in all samples. For S_2, small rod-shaped bacteria structures were well recovered in reconstructed HR images from LR images. The boundaries between cells are difficult to distinguish in the LR and interpolated HR images of S_4. Reconstructed HR images, however, show more explicit boundaries close to GT images. SRI-FLIMnet also has good performance in reconstructing correct lifetime distributions. The lifetime distributions of LR images shift towards a smaller lifetime due to the pixel binning of decay histograms among neighboring pixels. This problem is resolved in reconstructed HR images, which generate similar lifetimes with GT images. More apparently, the outer boundaries with a higher lifetime are also recovered. These results demonstrate that SRI-FLIMnet can reconstruct more useful HR images from their LR counterparts without prior knowledge.

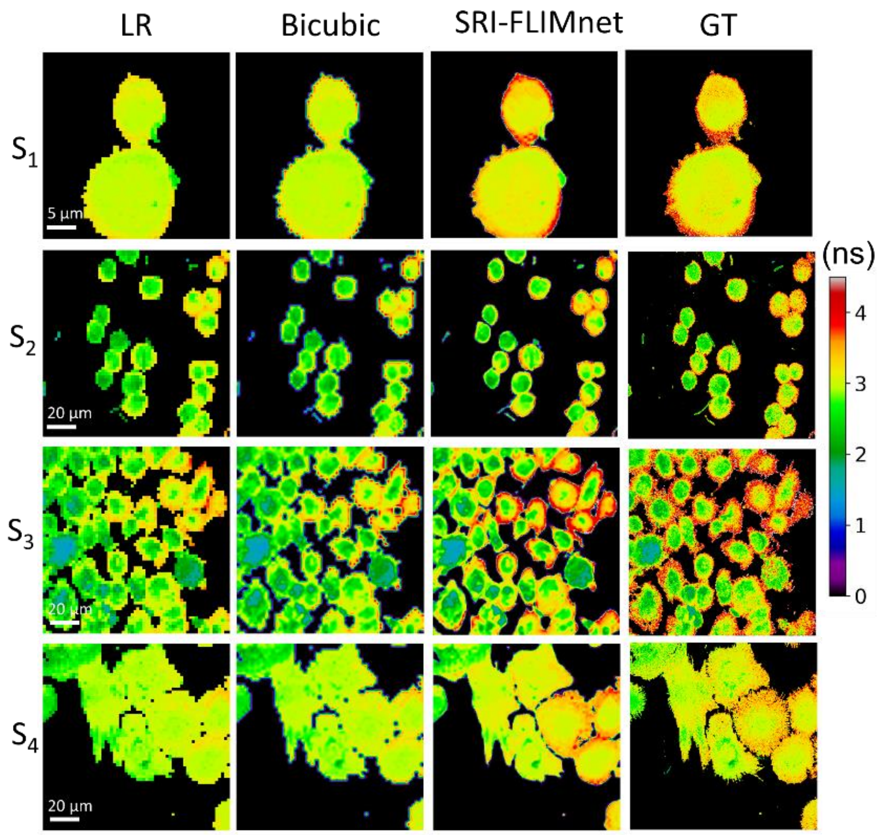


Fig. 4.7. Spatial resolution improved images of experimental FLIM data of bacteria-infected mouse raw macrophage cells. The 1st to 4th columns denote four different samples denoted by S_1 to S_4. The 1st row shows the normalized intensity images of the four samples. Their FL images are shown in the 2nd row. The 3rd row shows the reconstructed HR images by SRI-FLIMnet. The last row is the GT HR images for reference.

Fig. 4.8 shows the normalized absolute error maps and lifetime distribution histograms in Fig. 4.7. The results are similar to those in Fig. 4.6. Large discrepancy only occurs at boundaries. The lifetime distributions are also close to GT images. Figs 4.7 and 4.8 indicate that SRI-FLIMnet trained by semi-synthetic datasets can be well applied to new experimental datasets. The semi-synthetic datasets and the proposed SRI-FLIMnet are expected to find more DL-based FLIM applications. However, it should be noted that SRI-FLIMnet is limited to recovering very fine structures (high-frequency contents) and producing smooth images. This can be seen from the reconstructed images of S_1 to S_4 in the 3rd column, in which irregular and rough outer edges of cells are over-smoothed.

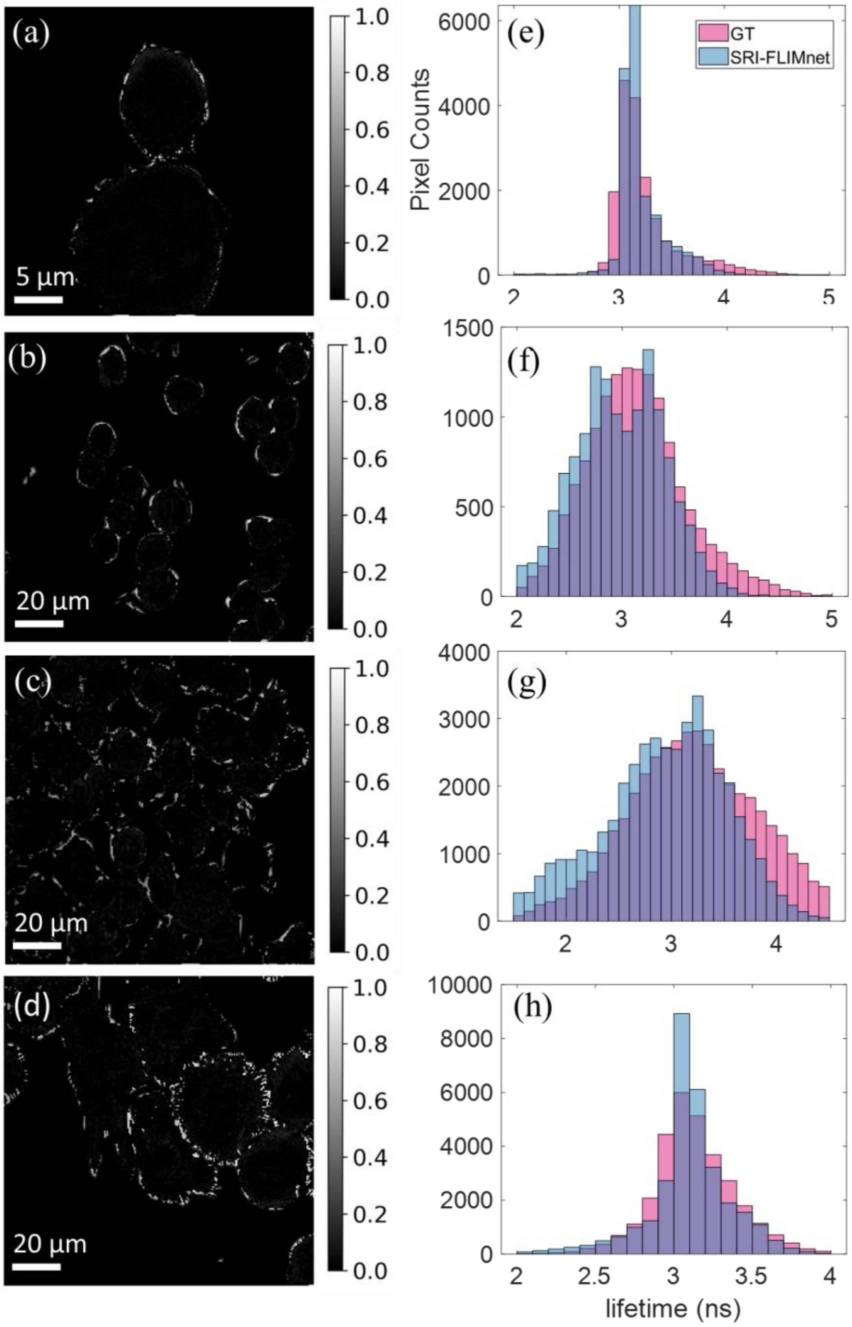


Fig. 4.8. Normalized absolute error maps (a) - (d) and lifetime distributions of the GT and SRI-FLIMnet reconstructed HR images (e) - (h) for experimental data S_1 – S_4. The normalized absolute error maps have an arbitrary unit.

4.6 Conclusion

In this work, we first developed a theoretical method to generate massive semi-synthetic FLIM data. It is challenging to use traditional FLIM experiments to image every cellular morphology and various fluorophores with an extensive range of lifetimes and complex decays. Therefore, there are no available large-scale FLIM datasets in scientific communities. Our work offers a feasible way to fill the gap and develop FLIM-related deep learning techniques. A degrading model was also proposed to obtain LR-HR image pairs with different scaling ratios, essential for spatial resolution enhancement tasks. We only considered a general case in the preparation of the training dataset. The semi-synthetic FLIM dataset can be further optimized according to specific tasks by considering cell types, fluorophores, SNR, and other noise sources to perform better. Our method is expected to significantly boost deep learning techniques for FLIM applications.

In addition to the semi-synthetic FLIM data preparation, a high-efficient neural network architecture, SRI-FLIMnet, was designed to map LR FLIM data to HR images. Achieving HR for FLIM is more challenging and complicated. Since FLIM is an indirect imaging technique, the lifetime should be evaluated from measured decay histograms. Meanwhile, LR FLIM data has significant information loss in spatial and temporal dimensions. To address these issues, SRI-FLIMnet adopts a combined strategy to estimate lifetimes and improve spatial resolution for LR images sequentially. Two subnets, τ-Net and SR-Net, are responsible for the two steps, respectively. This configuration is flexible to tailor the network for different situations. For example, it has been demonstrated that a narrower input lifetime range will provide higher estimation accuracy in τ-Net [152]. If the samples' lifetime range is known in advance, τ-Net can be fast re-trained within a minute to perform better. We used the average lifetime as the indicator in this work, but τ-Net can be further extended for multi-exponential decay analysis. The corresponding SRI-FLIMnet can improve the spatial resolution for individual lifetime parameters, which is useful for FLIM-FERT applications. Both semi-synthetic and experimental FLIM datasets were applied to show that SRI-FLIMnet can show more apparent boundaries and closer lifetime distributions to GT images. However, it should be noted that SRI-FLIMnet cannot precisely reconstruct images at the pixel level. Theoretically, mapping LR to HR images is an ill-posed problem, as LR images do not contain detailed information. Deep learning approaches can learn statistical features but cannot fully recover all information. Even when we use generative models like generative adversarial networks (GANs), the network would only generate plausible and artificial details, which may misrepresent original images. Therefore, a more dedicated image sampling method should be developed to preserve more information to provide stronger regularizations in DL tasks.

In summary, we propose a deep learning-based spatial resolution improved FLIM algorithm. The proposed SRI-FLIMnet trained by a semi-synthetic dataset shows superior performance in reconstructing HR images from LR 3D FLIM data. The performance is also verified by experimental data for bacterial-infected mouse raw macrophage cells. Our approach offers a practical computational method to fast obtain HR images without conducting extra experiments and modifying the system setting. SRI-FLIMnet has great potential for various FLIM-related applications.

Chapter 5

Fluorescence Lifetime Imaging with A Few Photons

5.1 Introduction

This chapter presents a DL algorithm that robustly estimates fluorescence lifetime under extreme low-light conditions, where only a few photon-per-pixel (PPP) are available. The need for a high photon budget to accurately quantify fluorescence lifetimes has long been a challenge in fluorescence lifetime imaging (FLIM), which has troubled the scientific community for some time. Although various methods have been proposed to analyze fluorescence lifetime, such as curve-fitting methods, phasor method, and emerging DL approaches, existing algorithms can only provide reliable results with sufficient photons. For instance, the widely used least-square fitting method requires at least a thousand photons for accurate lifetime estimation [65]. Maximum likelihood estimation and Bayesian analysis exhibit better performance for photon-starvation scenarios with several hundred photons [69-72], but at the cost of prolonged optimization, slow speed, and complex implementation. Recent advancements in DL approaches show promising results in analyzing low-light FLIM with several hundred photons [132, 152]. However, as the detected photons further decrease, all existing FLIM analysis methods become invalid. The reason is evident: Fig. 5.1 displays FLIM images with high and low PPP measured by TCSPC. Current analysis methods follow the convention of pixel-wise lifetime analysis, i.e., independently evaluating the fluorescence lifetime in a single pixel. If the pixel in the FLIM image has sufficient photons, the lifetime can be evaluated using a specific method. However, if there are only several or dozens of photons in the pixel, as depicted in Fig. 5.1 (c) and (d), although the intensity image still has a good SNR to keep the cell morphology (shape and structure), the decay information is permanently lost. Thus, it is mathematically impossible to recover the fluorescence lifetime within the pixel. For low-intensity FLIM images, pixel binning can be applied to accumulate sufficient photons for lifetime estimation. But it sacrifices spatial resolution. Alternatively, global analysis can also be used for lifetime analysis in this case. In the global analysis, all photons in the FLIM image are accumulated in one histogram to estimate the fluorescence lifetime[171, 172]. It is not suitable for FLIM images with spatially varying lifetimes.

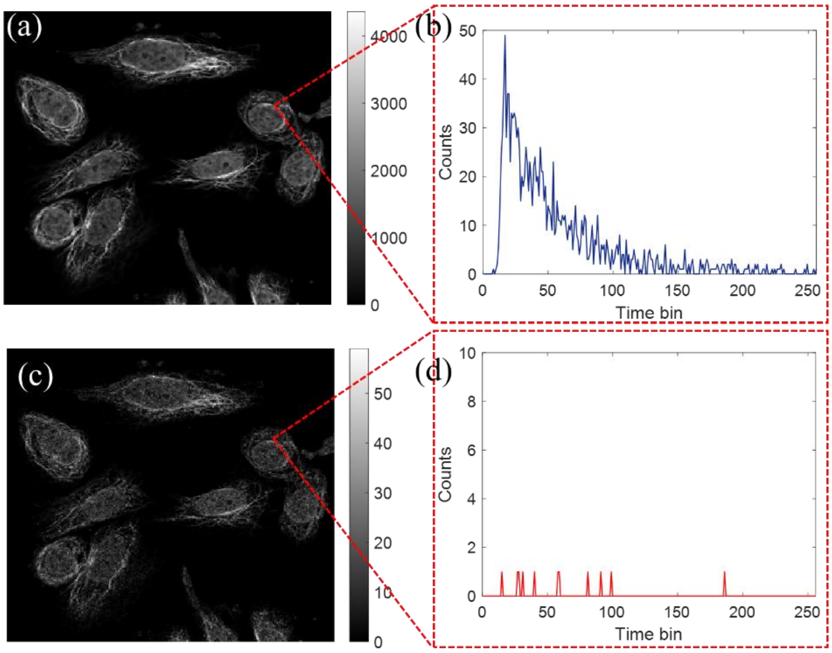


Fig. 5.1. FLIM images with high and low PPP measured by TCSPC. (a) and (c) FLIM images with several thousand and only a few PPP, respectively. (b) and (d) Decay histograms of a randomly selected pixel in the same position in (a) and (c), respectively.

Obtaining sufficient photons can be difficult or even impossible in many cases, which may be attributed to one or both factors below:

- **The available photon counts are limited.** The fluorophores of interest may have a low concentration, undesirable quantum yield, and photosensitivity in cells. Their excitation and emission wavelengths are unoptimized for the FLIM system. The low transfer or transporting efficiency becomes an important issue for exogenous fluorophores introduced by transferring fluorescent proteins and transporting fluorescent molecules.

- **Many FLIM measurements are strictly time-constraint.** For live-cell FLIM, the excitation laser pulses can cause destructive effects on cell vitality and cellular reproduction. It can also lead to photobleaching and cell lysis. If maintaining biological viability and avoiding photo-perturbation become a major concern, the measurement time should be as short as possible, and the laser power should remain low. For the investigation of dynamic biomedical phenomena, the molecular interactions of interest may have very fast kinetics, leading to a limited measurement window. In addition, for the characterization of high throughput cell samples in flow cytometry, the samples move through the laser interrogation point fast, and the measurement window is also narrow due to a short dwell time.

Despite the versatility of FLIM, its application range is largely limited by the available photons. In this chapter, we break the pixel-wise analysis paradigm and propose a deep learning approach to robustly analyze fluorescence lifetimes in extreme low-light conditions. Our algorithm, called Few-Photon Fluorescence Lifetime Imaging (FPFLI), takes advantage of two facts:

(a) The fluorescence lifetimes are spatially correlated among neighboring pixels.

(b) In TCSPC measurement, intensity images have a good SNR with few photons to preserve spatial information.

We hypothesize that the information in the intensity image can benefit the lifetime estimation in a FLIM image. FPFLI adopts a hybrid deep neural network architecture. A 1D network is dedicated to local lifetime estimation, and a 2D network is used to fuse the local fluorescence lifetimes and the spatial distribution of photon intensity. Our algorithm pushes the photon budget of FLIM to a recorded low level with a few photons per pixel. To our knowledge, this is the first low-count tool to offer such a robust performance.

5.2 Training dataset preparation

We adopt the framework of generating semi-synthetic FLIM datasets proposed in Chapter 4 to create low-count FLIM datasets for training.

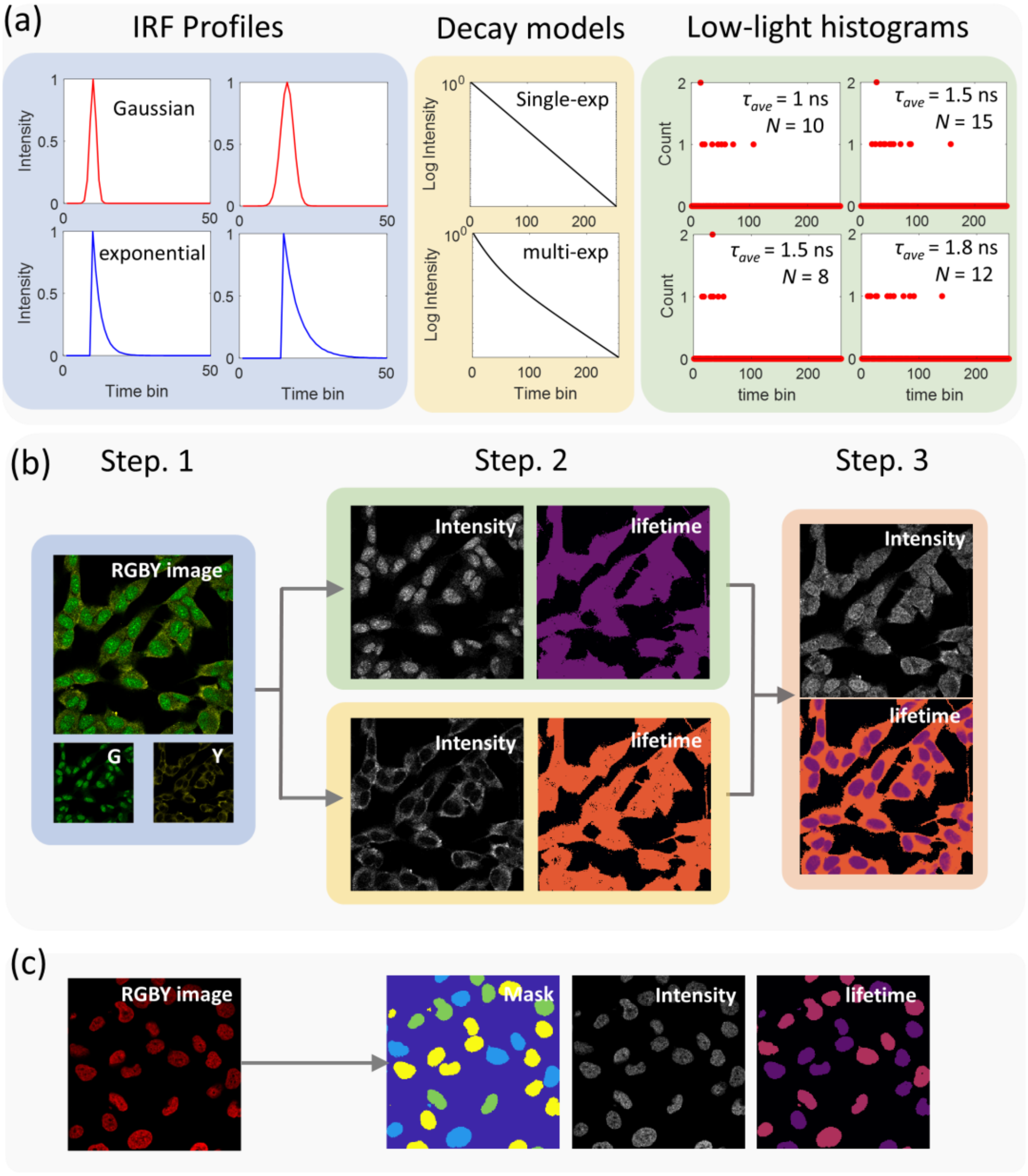


Fig. 5.2 Illustration of the generation of synthetic decays and semi-synthetic FLIM images. (a) The process of generating low-light synthetic decays. (b) The process of generating semi-synthetic FLIM images using cell images obtained by confocal immunofluorescence

microscopy systems in HPA datasets. The image has four color channels, containing nuclei (blue), microtubules (red), endoplasmic reticulums (yellow), and proteins, such as endosomes, lysosomes, or mitochondria (green). (c) The process of generating lifetime images using a single-color channel with multiple lifetimes.

The generation process of low-light FLIM images is illustrated in Fig. 5.2. Fig. 5.2 (a) shows the generation of low-light decays. First, we select the impulse response functions (IRFs), which are characterized by their profiles, peak positions, and full-width-at-half-maximum (FWHM). In our study, we considered two types of IRFs: Gaussian and exponential IRFs. These are good approximations to the IRFs of most Photomultipliers (PMTs) and Single-photon avalanche diode (SPAD) detectors, making them suitable for a wide range of FLIM systems [45]. Next, we choose single- or multi-exponential models to convolve with IRFs to generate decays. We limit our consideration to single-, bi-, and tri-exponential models since decays with more lifetime components can be represented equivalently by these models.

The low-count semi-synthetic FLIM images are generated from the HPA dataset. Fig. 5.2 (b) illustrates the detailed process to generate FLIM images. To generate FLIM data, we first randomly select channels from the original images, such as the Green (G) and Yellow (Y) channels. The second step involves scaling the intensity image, assigning lifetime values to channels, and generating decays for each pixel. Consider an intensity image $X_{ij} \in [0, 255]$ in one channel, where *i* and *j* denote the pixel positions in vertical and horizontal dimensions. We linearly scale X_{ij} by dividing a scaling factor *F*, where *F* is calculated as:

$$F = M/r, \tag{5.1}$$

Here, *M* is the mean value of valid pixels in the original intensity image, where valid pixels are those with an intensity greater than 20 to exclude background pixels. *A* is a random number ranging from 5 to 10. The scaling ensures that the average intensity of the image equals *A*. Assume the intensity at pixel (*i*, *j*) is Y_{ij}, Y_{ij} follows a Poisson distribution as:

$$P(Y_{ij} | \lambda_{ij}^{s}) = \frac{(\lambda_{ij}^{s})^{Y_{ij}}}{Y_{ij}!} e^{-\lambda_{ij}^{s}}, \tag{5.2}$$

where $\lambda_{ij}^{s} = \boldsymbol{round}\ (X_{ij}/F)$ is the rate parameter of the Poisson distribution. ***round***() is the operator to choose the input value to its nearest integer. After obtaining Y_{ij} from Eq. (5.2), we calculate the temporal distribution of photon arrival time using the forward TCSPC model, which has been described in section 4.2.1. An average lifetime from 1 to 4 ns is randomly assigned to each color channel to generate decays, which can be single or multi-exponential. In the third step, we sum up all selected channels. The lifetimes in the GT lifetime image are the average lifetime of all lifetime components. Since different color channels have diverse intensities and spatial overlaps, the semi-synthetic FLIM images naturally show complex decay features and wide lifetime distributions. We also considered a single-color channel with multiple lifetimes in different parts to increase the diversity of spatial distribution. As illustrated in Fig. 5.2 (c), the original intensity image is segmented into different groups, and each group is assigned a different lifetime to generate synthetic FLIM images.

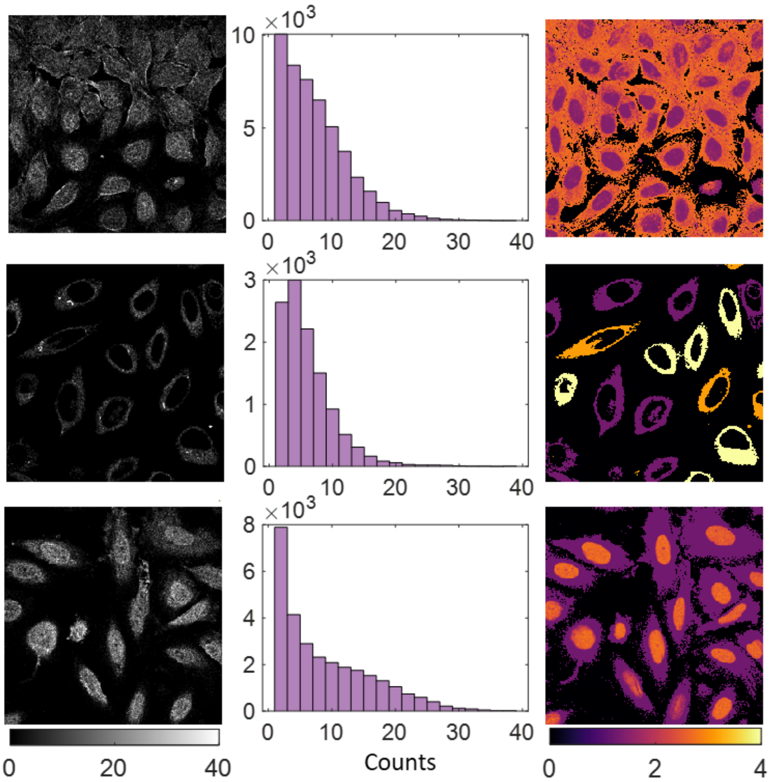


Fig. 5.3. Samples of low-count FLIM images for network training. The first column on the left side is the intensity images. The second column shows the intensity distributions of the samples. Most pixels contain less than ten photons, and the photon count has a large range.

The PPP of the samples ranges from 5 to 10. The last column is the ground truths of the samples' lifetime distributions.

Fig. 5.3 shows samples of generated low-count FLIM images. The first column shows the intensity maps of the FLIM images. The intensities have a wide distribution from zero to dozens, and the PPP remains a low value, only a few counts. The second column shows the histogram of the intensity distribution. Most pixels only have several photons. The third column shows the ground truth lifetime maps, which are the amplitude-averaged lifetimes of all considered channels. The FLIM images with a few PPP have a wide lifetime range, various spatial distributions, and complex temporal decays. Any traditional method cannot be used to estimate the fluorescence lifetime. A massive dataset with more than 10,000 samples is generated for algorithm training.

5.3 Algorithm implementation

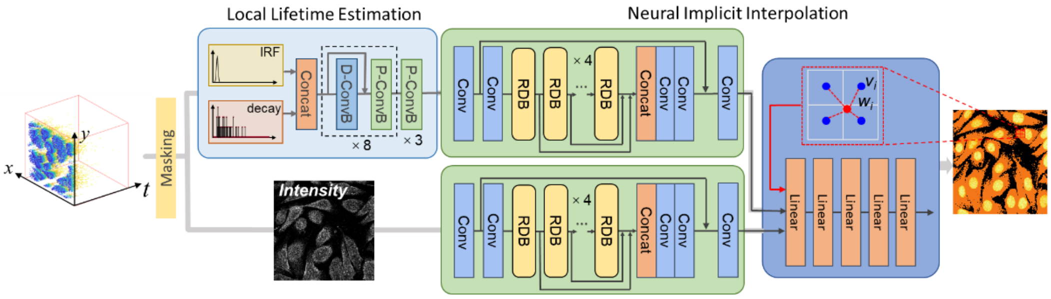


Fig. 5.4. Overview of FPFLI algorithm. A mask first removes the background noise pixels of the input 3D FLIM data. A local lifetime estimator composed of depth- and pointwise-convolution blocks (D- and P-ConvB) estimates the local lifetime map through pixel-binning. Two residual dense networks encode the local lifetime map and intensity image to the local implicit image functions. Then the final fluorescence lifetime image is reconstructed through neural implicit interpolation using a dense neural network. Detailed architectures and parameters will be explained in later content.

Fig. 5.4 shows the FPFLI algorithm overview for estimating fluorescence lifetime from extremely low-intensity FLIM data. FPFLI contains three steps: imaging masking, local lifetime estimation, and neural implicit image interpolation (NIII). Below are the explanations of the three steps.

5.3.1 image masking

Our computational framework contains three steps to evaluate fluorescence lifetime images from measured data with a few PPPs. Assume the input FLIM data is $\boldsymbol{H}(x, y, t)$, where $x, y \in [1, M]$ are the two spatial coordinates and $t \in [1, N]$ is the temporal coordinate. The first step is image masking to filter out background noise pixels and retain sample morphology. In TCSPC FLIM measurement, the Poisson noise originating from the photon detection process plays a dominant role. When detected PPP is much larger than 1 (e.g., PPP = 5), the background noisy pixels can be easily distinguished according to their photon counts. Therefore, the data masking becomes simpler by applying a proper lower threshold. The mask **X** is defined as:

$$\boldsymbol{X}(x, y) = \begin{cases} 0, \boldsymbol{I}(x, y) \leq threshold \\ 1, \boldsymbol{I}(x, y) > threshold \end{cases}' \tag{5.3}$$

However, when the detected photons are extremely low (e.g., about 1 PPP), the background noise arising from the dark count noise of sensors cannot be neglected any longer. In this case, the noise pixels would severally degrade sample structures. So, they should be removed to reconstruct the sample morphology and eliminate the effect on lifetime determination at the initial stage. According to the empirical fact that the detected fluorescence signals have a strong spatial correlation, we generate the mask using Markov random field (MRF) model [173]. Firstly, the measured intensity image obtained by integrating the temporal dimension is binarised by assigning +1 and -1 to pixels with and without photon detection, respectively. The binary intensity image is denoted by $I_{ij}^{b} \in \{-1, +1\}$ with $i, j \in [1, M]$. In the MRF model, I_{ij}^{b} is assumed to be a clear image $I_{ij}^{c} \in \{-1, +1\}$ with $i, j \in [1, M]$ contaminated by noise. The clear image I_{ij}^{c} is obtained by minimising the energy function:

$$minimizing\ E(\boldsymbol{I^c}, \boldsymbol{I^b}) = \sum_{i=1}^{M} \sum_{j=1}^{M} \left[\alpha I_{ij}^{c} - \beta \sum_{i', j' \in N_b(i,j)} I_{ij}^{c} I_{i'j'}^{c} - \eta I_{ij}^{c} I_{ij}^{b} \right], \tag{5.4}$$
$$subject\ to\ I_{ij}^{c} \in \{-1, +1\}, \forall i, j \in [1, M]$$

where N_b (i, j) denotes the 4-pixel neighborhood of pixel (i, j). α, β, η are non-negative parameters. Various algorithms can be applied to resolve the above optimization problem, such as iterated conditional models (ICM), graph-cut algorithms, and MCMC [174]. In this work, the ICM was used because it is fast and effective. The parameters are set as $\alpha = 1$, $\beta = 1.4$, and $\eta = 2$. Once $\boldsymbol{I}^c$ is obtained, the mask $\mathbf{X}$ is calculated by a simple affine transformation as:

$$\boldsymbol{X} = (\boldsymbol{I^c} + \mathbf{1})/2, \tag{5.5}$$

The masked image is obtained by:

$$\boldsymbol{H}' = \boldsymbol{X} \cdot \boldsymbol{H}, \tag{5.6}$$

The operator · denotes the dot product (scalar product) of the matrices. The intensity image is $\boldsymbol{I}(x,y) = \sum_{t=1}^{N} \boldsymbol{H}'(x,y,t)$.

5.3.2 *Local lifetime estimation*

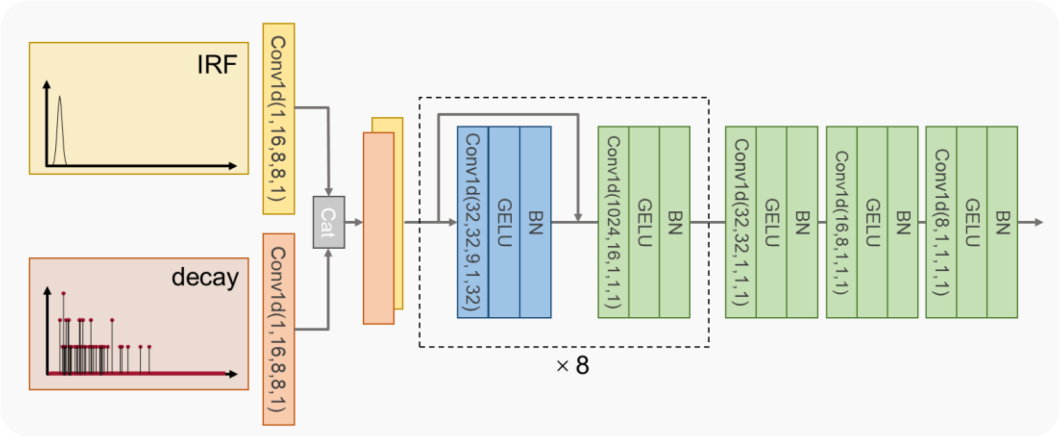


Fig. 5.5. Detailed architecture of the local lifetime estimator. The layer parameters of 1D convolution layers are indicated as (input channel, output channel, kernel size, stride, group).

The second step is the local lifetime estimation. The neighboring pixels are integrated through spatial pixel binning to utilize the spatial correlation property of FLIM images. For example, an 8×8-pixel patch or an even larger 16×16-pixel patch is merged into one pixel, in which the photon shall accumulate up to tens to hundreds of photon count. Hence, the local fluorescence lifetime estimation becomes well-posed. The FLIM histogram with a low spatial resolution after spatial pixel binning is calculated using Eq. (4.5):

$$\boldsymbol{H}^l = 1^{k\times k\times 1} \otimes \boldsymbol{H}', \tag{5.7}$$

where k is the binning factor, $\otimes_{s=(k,k,1)}$ is the three-dimensional cross-correlation operator with the stride $s = (k, k, 1)$ along the spatial and temporal dimensions. A local lifetime estimator (LLE) is used to evaluate the binning area's average lifetime. To robustly calculate fluorescence lifetimes under such a low-photon condition, we redesigned the 1D-CNN algorithm proposed in Chapter 3. Fig. 5.5 shows the detailed network architecture of LLE. LLE adopts the ConvMixer architecture as the backbone, which features patches as input representation, isotropic architecture with equal size and shape, and separated depth and pointwise convolutions [175]. In our previous study, the DL algorithm was designed for a specific FLIM system, and we assumed the IRF was fixed for all input samples. However, when data are acquired from different systems, the algorithm's accuracy significantly degrades since the shape and position of the IRF may differ from those of the predefined IRF in the training phase. To address this problem, LLE considers IRF and decay, broadening its application range to different FLIM systems. The decay and its corresponding IRF are fed into the network and divided into identical patches by two independent patch embedding layers. A convolution layer can serve as the patch embedding layer. The convolution layer is set as h output channels, p kernel size, and p stride to obtain patch embeddings with patch size p and embedding dimension h. In our study, we set p = 16 and h = 8. Then the patch embeddings concatenate into one tensor before they are passed to eight identical ConvMixer blocks. Each block contains depth-wise and pointwise convolution layers with their connected activation layer (GELU) and batch normalization layer. The depth-wise and pointwise convolution layers efficiently extract the decay features by mixing channel-wise and spatial-wise information, respectively. After processing by ConvMixer blocks, three pointwise convolution layers converge the information and output the lifetime values. It should be noted that LLE has a much deeper architecture than the 1D CNN algorithm proposed in Chapter 3. A general tendency in DL is "the deeper architecture, the better performance" because deeper architecture can effectively capture high-dimensional abstract features. However, deeper architecture usually causes a mass of network parameters and difficulty in network training.

Thanks to the advanced network design. LLE has much fewer parameters compared to previous DL FLIM algorithms. The total parameter of LLE is only 28,865, which is only 59% of 1D CNN, and 2.5% of FLINET [132, 152].

LLE is trained by synthetic decays, as introduced in Chapter 3. The lifetime and photon count range from 1~4 ns and 50~1000 counts, respectively. The lifetime components are randomly set to 1 to 3. Two types of IRF are considered: Gaussian and exponential IRFs with FWHM from 10 to 40 ps and peak position from 10 to 30. The IRF configuration covers general TCSPC systems. The number of total training samples is 100,000. The loss function is the L_2 norm, and the ground truth target is the amplitude-weighted average lifetime to avoid the model mismatch problem [170].

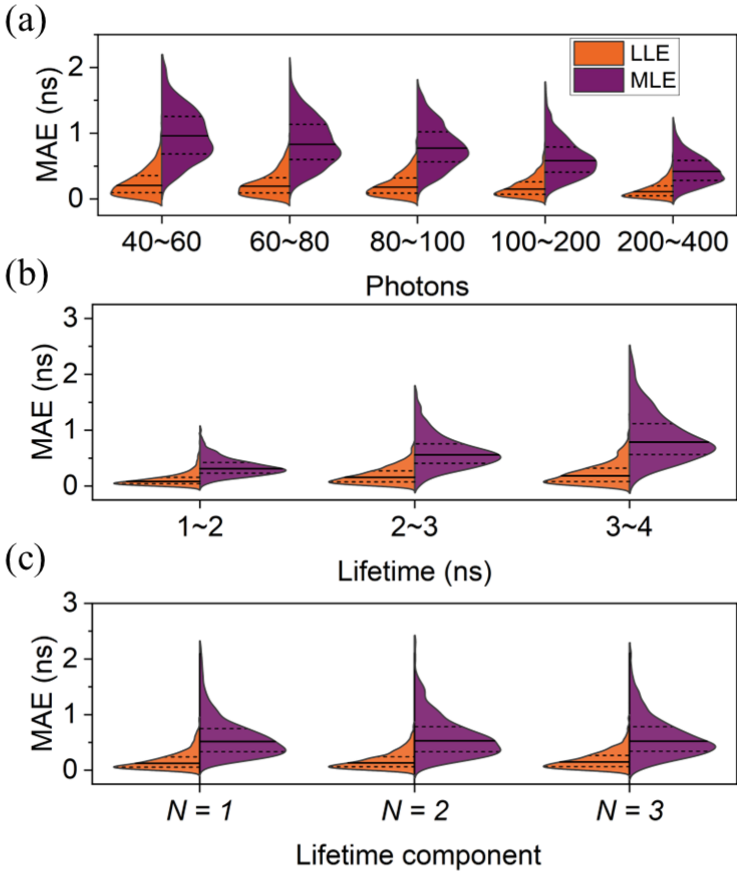


Fig. 5.6. Violin plots of MAE for LLE and MLE. MAE for LLE and MLE as a function of (a) photon counts, (b) lifetime, and (c) lifetime components. The photons, lifetime, and lifetime components ranges are 40~400, 1~4 ns, and 1~3. When one variable is under

investigation, other variables change in their corresponding ranges randomly. The solid line indicates the median value, and the two dashed lines indicate the upper and lower quartiles.

Fig. 5.6 illustrates the performance of LLE in lifetime estimation under low-light conditions, using the mean absolute error (MAE) as a figure-of-merit for evaluating algorithm accuracy. Results for the maximum likelihood estimation (MLE) algorithm are also presented for comparison. We use MLE as a benchmark because MLE is one of the most widely used algorithm for robust low-light FLIM data analysis. Fig. 5.6 (a) displays the MAEs for LLE and MLE as photon counts vary from 40 to 400. The median MAE for LLE is significantly smaller, with a narrower distribution than that of MLE, indicating superior accuracy of LLE in low SNR conditions. For instance, when photons range between 40 and 60, the median MAE for LLE is 0.2 ns, with corresponding quartiles of 0.09 and 0.36 ns, respectively, surpassing previous algorithms by a large margin. Additionally, the MAEs for LLE in Fig. 5.6 (b) and (c) show a considerable reduction in error, indicating robust performance for a wide range of lifetimes and different lifetime component numbers. Overall, LLE demonstrates a record-breaking performance in lifetime estimation. We denote LLE as $C(\boldsymbol{\theta};\cdot)$ with parameters $\boldsymbol{\theta}$. The local lifetime image $\boldsymbol{L^l}$ is calculated as:

$$\boldsymbol{L^l} = C\left(\boldsymbol{\theta}; \boldsymbol{H^l}\right), \tag{5.8}$$

5.3.3 Neural implicit image interpolation

The third step uses NIII to reconstruct the fluorescence lifetime image based on the local fluorescence lifetime image $\boldsymbol{L^l}$ obtained in step 2 and the intensity image $\boldsymbol{I}$. To conduct NIII on two images with different spatial resolutions, implicit neural representations (INRs) were used to parameterize both images. INRs are a novel approach for signal parameterization, employing a continuous and differentiable implicit function to map coordinates to the corresponding signal, regardless of the signal's dimensionality (e.g., 1D audio signals, 2D images, 3D point clouds, etc.) [176-180]. Unlike analytical functions, INRs have no explicit form and are parameterized by a deep neural network like Multi-Layer Perceptrons (MLP). INRs effectively capture signals' underlying features using fewer parameters and are independent of spatial resolutions, enabling them to be sampled at arbitrary resolutions. Encoder-based methods are proposed to share knowledge of all inputs, rather than fitting individual functions for each instance. An encoder extracts latent codes from various inputs, and a decoder takes these codes and coordinates to reconstruct the input signal. Encoder-based methods have been successful in numerous 3D tasks. Local Implicit Image Function (LIIF), for example, extracts a set of latent codes distributed in spatial dimensions to represent natural images [176]. The decoder queries the local latent codes around the coordinate for a given coordinate and predicts RGB values.

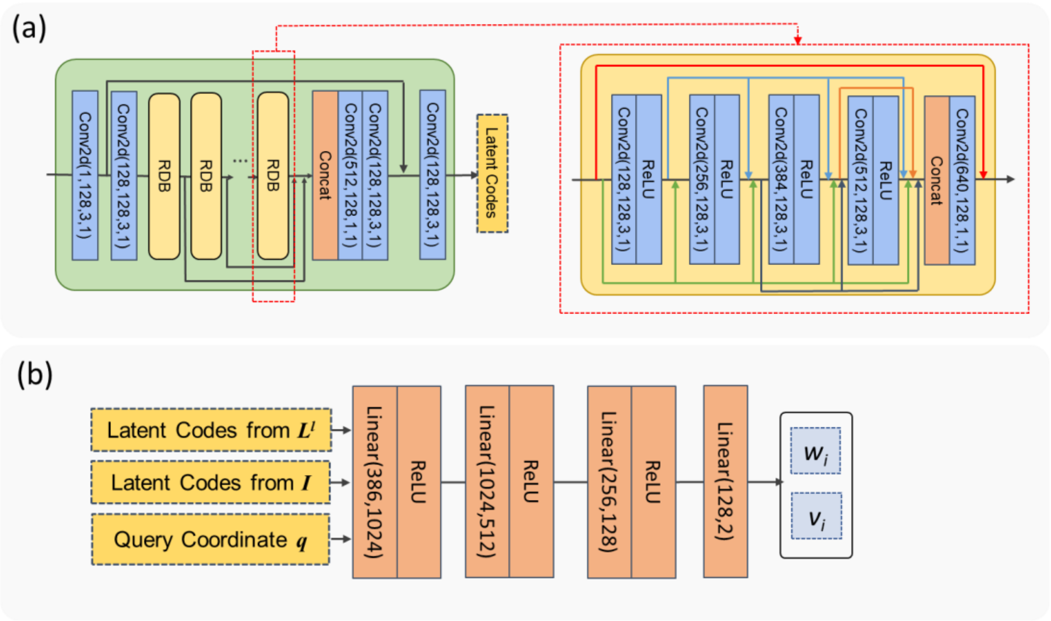


Fig. 5.7. Detailed encoder and decoder architecture for parameterizing the image with implicit neural representations. The encoder and decoder use a residual dense network and dense neural network as backbones, respectively.

To decouple the spatial resolutions of two images in our study, we parameterized them using INR and extracted two sets of latent codes from both images using identical encoder networks. Fig. 5.7 (a) shows the detailed structure of the encoder networks used. We adopted the residual dense network (RDN) as the backbone to learn local and global hierarchical features and extract latent codes, which was discussed in detail in Chapter 4. The RDN encoder consists of four identical residual dense blocks (RDB) to learn local features with different levels. The insert of Fig. 5.7 (a) illustrates the architecture of the RDB, with four densely connected 2D convolutional layers. In RDB, the first layer's input feature is 128, and the feature number's growth rate in the subsequent layers is also 128. A concatenation layer is added at the end of the RDB, followed by a residual connection across the entire block. Two independent and identical encoders are applied to extract two sets of latent codes from $\boldsymbol{L}^l$ and $\boldsymbol{I}$:

$$\begin{cases} z_i = E\left(\boldsymbol{\phi}; \boldsymbol{L}^{\boldsymbol{l}}\right)(x_i) \\ g_j = E(\boldsymbol{\varphi}; \boldsymbol{I})\left(x_j\right) \end{cases}, \tag{5.9}$$

where z_i and g_j are the latent codes from $\boldsymbol{L}^l$ and $\boldsymbol{I}$ for the query pixel coordinates x_i and x_j, respectively (x_i and x_j denote the coordinate vectors for simplification). $E(\boldsymbol{\phi}; \cdot)$ and $E(\boldsymbol{\varphi}; \cdot)$ are the encoder with parameters $\boldsymbol{\phi}$ and $\boldsymbol{\varphi}$, respectively.

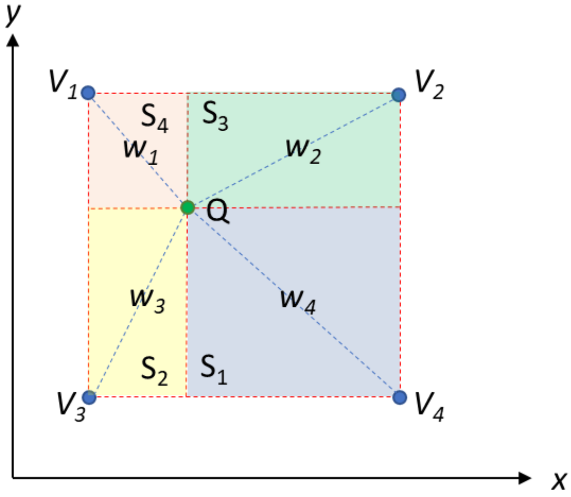


Fig. 5.8. Schematic view of image interpolation.

To better illustrate NIII, we first review the traditional interpolation method used in image processing to resample images. As shown in Fig. 5.8, to obtain the value of a given query point Q for interpolation, the first step is to find the nearest neighboring vertices, usually the four nearest corner pixels V_i, i =1, 2, 3, 4. Then the value of Q is the weighted average of the four pixels:

$$q = \sum_{i=1}^{4} w_i \, v_i, \tag{5.10}$$

where w_i and v_i are the corresponding weighting coefficient and the pixel value of V_i. Different interpolation methods use different weighting coefficients. Take the most widely used bilinear interpolation as an example. The weighting coefficient is calculated as follows:

$$w_i = \frac{S_i}{S}, \tag{5.11}$$

where S_i is the partial area diagonally opposite the corner V_i. $S = \sum S_i$ is the total area.

In NIII, we deviate from the traditional approach of directly using vertex values from input images with specified weighting coefficients. Instead, we use a deep neural network (decoder) to generate these values from the latent codes and query coordinates [176, 181]. Given a query pixel coordinate x_q and two sets of latent codes z and g, the weight and vertice values of the neighbor pixels are calculated by a DNN $F(\boldsymbol{\sigma},)$ with parameters $\boldsymbol{\sigma}$:

$$a_{q,i}, v_{q,i} = F\big(\boldsymbol{\theta}; z_i, g_i, g_q - g_i, x_q - x_i\big). \tag{5.12}$$

Then the normalized weights are obtained by applying the SoftMax function:

$$w_{q,i} = \frac{e^{a_{q},i}}{\sum_{i \in \boldsymbol{\mathcal{N}}_q} e^{a_{q,i}}}, \tag{5.13}$$

where $\boldsymbol{\mathcal{N}}_q$ is the set of neighbor pixels for q. Finally, the target lifetime image $\boldsymbol{L}$ is calculated as:

$$\boldsymbol{L}(x_q) = \sum_{i \in \boldsymbol{\mathcal{N}}_q} w_{q,i} \, v_{q,i}. \tag{5.14}$$

The decoder is a five-layer FNN, as shown in Fig. 5.7 (b). It takes the two sets of latent codes and the query pixel coordinates as input, and the output is the vertices' values and their corresponding weights.

For the training of NIII, we input the LR local lifetime image and the intensity image into the NII network. Then we sample pixels from the GT lifetime image and query the decoder with the pixel's corresponding coordinates to predict the pixel values. The L_1 loss is applied for network optimization.

5.4 Algorithm Evaluation

5.4.1 Synthetic FLIM images

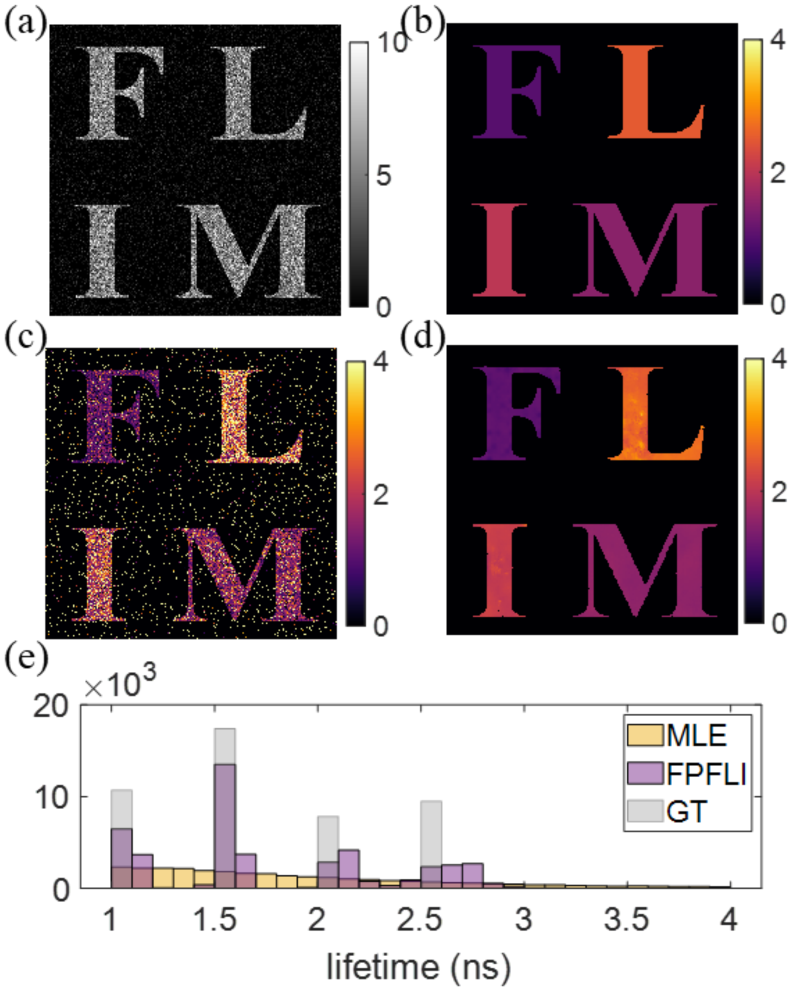


Fig. 5.9. FPFIL for analyzing synthetic low-light alphabetical FLIM image. (a) Intensity image. (b) Ground truth lifetime image. (c) and (d) Estimated fluorescence lifetime images using MLE (c) and FPFLI (d). (e) Histograms of fluorescence lifetime images for MLE, FPFIL, and GT.

We evaluated the effectiveness and robustness of FPFLI by applying it to a synthetic 512×512 FLIM image containing four capital letters (“F”, “L”, “I”, and “M”). Fig. 5.9 (a) and (b) show the corresponding intensity and lifetime images, respectively. The average PPP for the alphabetical FLIM image is approximately 5, and a background noise level of 0.33, modeled by a uniform distribution in the temporal dimension, was added to simulate dark count noise. Four lifetimes increasing from 1 to 2.5 ns with a 0.5 ns interval are assigned to the four letters. Fig. 5.9 (c) and (d) show the estimated lifetime images by MLE and FPFLI. MLE takes more than 10 mins to calculate the lifetime maps. However, it fails to calculate the lifetime as the photons are too fewer. Although we can observe faint differences among the four letters, it is impossible to obtain their lifetimes accurately. In contrast, FPFLI delivers a much better result within only 2 seconds, and the estimated lifetime image is very close to the ground truth. The background is clear as the noise pixels are masked. Fig. 5.9 (e) shows the lifetime histograms for MLE, FPFLI, and GT to better compare the two algorithms. The histogram for MLE has a flat distribution across the whole dynamic range from 1 to 4 ns. This is consistent with the observation in Fig. 5.9 (c). As for FPFLI, the histogram has four distinct clusters corresponding to the four lifetimes. The pixels with shorter lifetimes tend to have better accuracy. Most pixels with 1 and 1.5 ns lifetimes have exact lifetimes as the GT, and a small portion has a 0.1 ns deviation. However, FPFLI’s prediction accuracy becomes slightly worse for pixels with larger lifetimes because their predicted lifetimes have broader distributions. Nevertheless, the lifetime deviation is still within 0.2 ns, indicating FPFLI has superior performance in lifetime estimation. It is worth noting that we use FLIM data of cell images to train FPFLI. The morphologic features in the alphabetical FLIM image, such as linear boundaries and sharp corners, are rare in the training data. However, FPFLI can still fuse the intensity and local lifetime information. It demonstrates that FPFLI is a robust and effective method for analyzing various FLIM images.

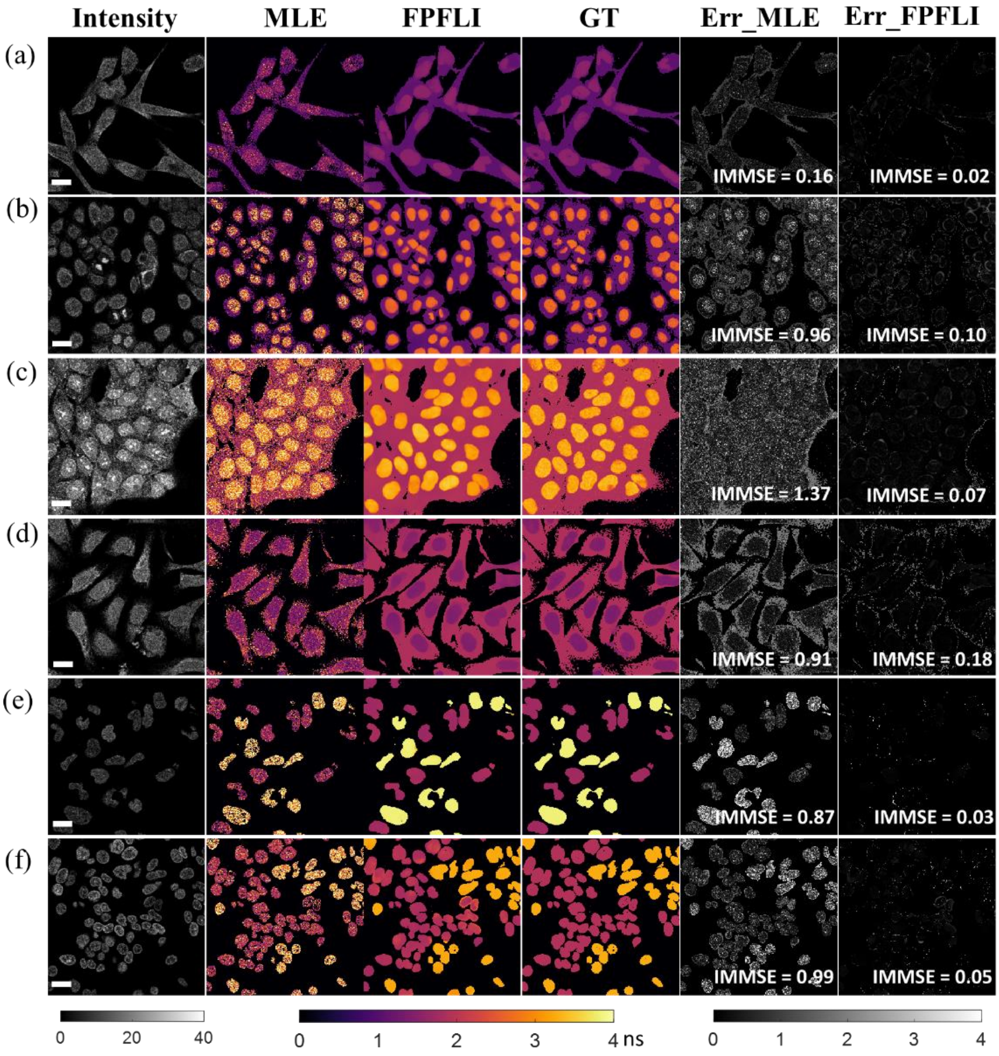


Fig. 5.10. FPFLI for analyzing semi-synthetic low-light FLIM image. The images' size is 256×256, and their average PPPs are less than 10.

We conducted further investigation into the performance of lifetime estimation on test semi-synthetic FLIM images. These images feature a variety of cell types and shapes and are unseen by FPFLI. Their fluorescence lifetimes have complex distributions with a large dynamic range, and their intensities range from 0 to 40, with average PPPs of less than 10. Fig. 5.10 displays six samples, showcasing their intensity, estimated lifetime, and ground truth lifetime images. Compared to conventional MLE, FPFLI achieves astonishingly accurate lifetime images, nearly identical to GTs. FPFLI can estimate complex local fluorescence lifetime distributions from various cellular organelles, even when the information is vague and indistinguishable in the intensity images. For example, in Fig. 5.10 (a), (b), and (d), the shapes of cell nuclei cannot be identified by human eyes, yet FPFLI can still reconstruct the nuclei's lifetime and shapes. In comparison, MLE only roughly distinguishes the lifetime difference in organelles, and we cannot calculate accurate lifetimes. To provide quantitative analysis, the error maps for MLE and FPFLI are shown in the last two columns. The error map is defined as $\left|\hat{\boldsymbol{L}} - \boldsymbol{L}^{GT}\right|$ with $\hat{\boldsymbol{L}}$ the predicted lifetime images by MLE or FPFLI. The error map for MLE has large and random errors, indicating that the lifetime estimation is an ill-posed problem, and MLE cannot converge to the correct lifetime. On the other hand, the error map for FPFLI has more minor errors, mainly occurring at the boundaries. The errors at the boundaries are well understood because the photons near the boundaries are often scarce and are neglected during the calculation. Additionally, we also use the IMMSE as a metric to quantitatively compare the two algorithms. IMMSE calculates the MSE between $\hat{\boldsymbol{L}}$ and $\boldsymbol{L}^{GT}$. IMMSE for FPFLI is much smaller for all samples, demonstrating that FPFLI is an effective algorithm to calculate fluorescence lifetime for such low light conditions.

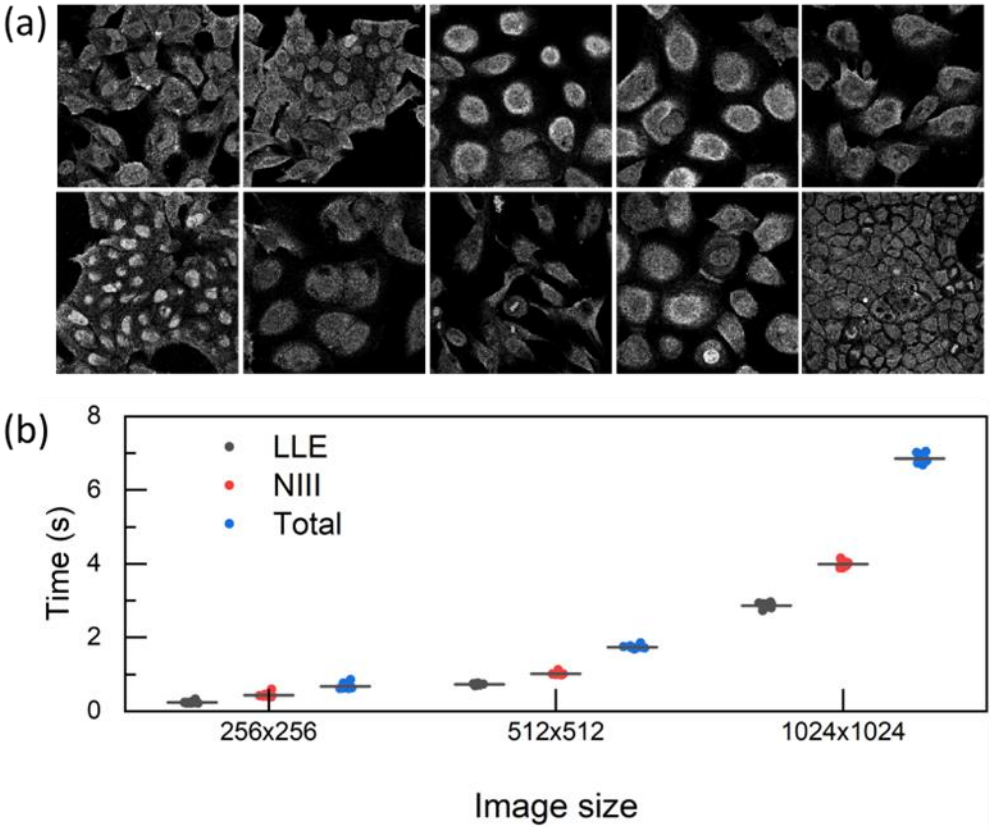


Fig. 5.11 Comparison of computation time on FLIM images with different sizes. (a) The

randomly selected images to generate FLIM images with 256×256, 512×512, and 1024×1024 pixels. (b) The computation time of LLE, NIII, and total process for three different sizes of FLIM images.

One distinct feature of FPFLI is the fast calculation speed owing to the unique lifetime reconstruction approach. Fig. 5.11 shows the computation time of FLIM images with different sizes. Ten images with varying granularity were randomly selected to create FLIM images with sizes of 256×256, 512×512, and 1024×1024 pixels, as depicted in Fig 5.11 (a). Fig. 5.11 (b) shows the computation time for different image sizes. For each image size, the computation time for all images was found to be similar because FPFLI has the same number of operations for these images. FPFLI has a very fast computation speed as it only needs to estimate a small number of local lifetimes, and the lifetimes in the remaining pixels can be obtained through interpolation. The mean time required to analyze 256×256 and 512×512 images was found to be only 0.63 and 1.72 seconds, respectively. Even for large-scale images, such as 1024×1024 pixels, the average time required was only 7.04 seconds. In comparison, MLE takes more than an hour to estimate lifetimes for 256×256 images.

5.4.2 FPFLI analysis of mouse kidney section

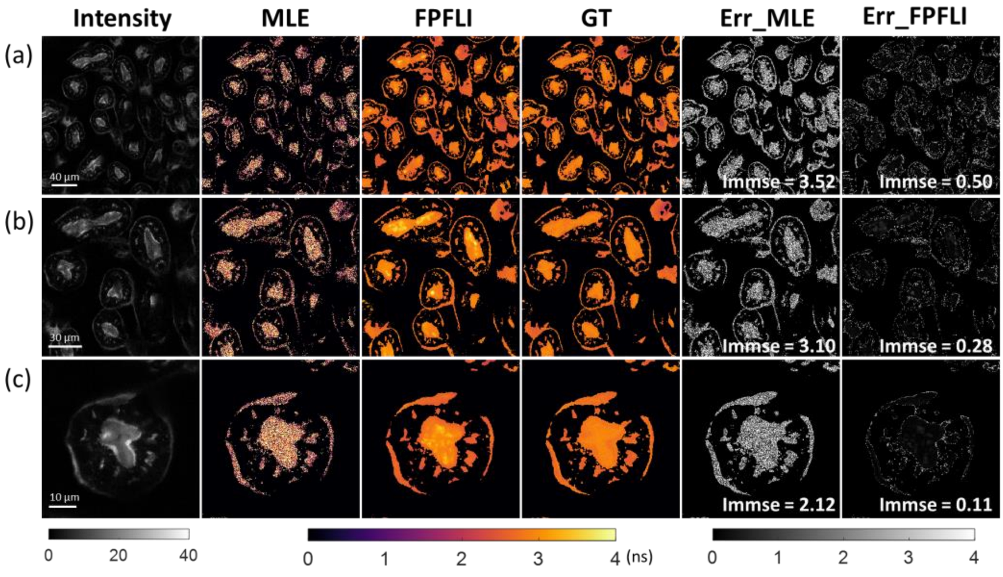


Fig. 5.12 FPFLI analyzing FLIM images of the mouse kidney section with different FOVs.

We tested FPFLI on experimental FLIM data obtained from a cryostat section of a mouse kidney stained with Alexa Fluor 488 wheat germ agglutinin (F24630, FluoCells™ Prepared Slide, Thermo Fisher, UK). We used a confocal microscopic FLIM system (MicroTime 200D, Picoquant, Germany) with a 483 nm picosecond diode laser operating at 40 MHz to measure the cell images. Fluorescence emissions were collected by a 63× water immersion objective and recorded by a single photon avalanche diode (SPAD) detector with a 530/30 nm bandpass filter. To obtain low-intensity images, we terminated the measurement when the maximum photon count reached 40. We also measured corresponding high-intensity images with a maximum photon count of 4000 and fitted them by MLE as GT images. The measured FLIM data has 512×512 pixels, and their histograms have a 20 picosecond (ps) timing resolution with a period of 50 ns. To reduce the time bins, we post-processed the images to 512×512×256 with a 0.1 ns timing resolution through temporal binning. Three cell images with different field-of-views (FOVs) are shown in Fig. 5.12, and their PPPs are around 6 to 7. The sample shapes and decay profiles show vast differences from the training data. We adapted FPFLI by retaining LLE using training data with a 0.1 ns timing resolution to analyze better the experimental data, which took only 3 minutes. The synthetic IRF was used because the SPAD detector has a wavelength-dependent response.

Consistent with previous findings, only FPFLI can produce meaningful lifetime images under the highly low-light conditions considered in this study. The overall lifetime distributions for all FOVs analyzed were in good agreement with their corresponding GTs, and subtle lifetime differences in some local areas were easily distinguishable. Quantitative analysis of IMMSE, similar to that for synthetic images, indicates that FPFLI trained by synthetic FLIM data has similar performance on different experimental FLIM data. The quick training of LLE enables FPFLI to adapt to varying decay histograms, making it a promising tool for general FLIM systems. However, it is worth noting that FPFLI can generate artifacts in some areas, such as the bright spots observed in the reconstructed lifetime images in Fig. 5.12 (a)-(c). These artifacts are mainly due to relatively large lifetime deviations resulting from low photon counts, which can be eliminated by increasing the detected photons. In addition, images with high spatial complexity, such as those with complex shapes or lifetime distributions, have larger IMMSEs, indicating that FPFLI is more challenging to reconstruct lifetime images in such cases. Nevertheless, our results experimentally demonstrate that FPFLI can robustly estimate lifetime images from extreme low-light conditions with only a few PPPs, which is impossible for conventional FLIM analysis methods.

5.4.3 Moving fluorophore-tagged microbeads

We then use FPFLI to analyse the experimental FLIM images of moving fluorophore-tagged microbeads in an aqueous solution. The samples were prepared by mixing two kinds of microbeads, crimson fluorescent microspheres (F8831, FluoSpheres™ Polystyrene Microspheres, Thermo Fisher, UK) and yellow-green (YG) fluorescent microspheres (F8836, FluoSpheres™ Polystyrene Microspheres, Thermo Fisher, UK). The average size of both microbeads is around 10 μm and the solution is 3.6×10^6 beads⁄mL. The solution was dropped on a glass slide and then covered by a coverslip. The microbeads randomly move due to the Brownian motion. We use a two-photon FLIM system described in Chapter 3 to acquire the FLIM images. The laser excitation wavelength is 800 nm, and the laser pulse width is less than 200 fs with an 80MHz repetition rate. The emission light was collected by a 60× water-immersion objective lens (N.A =1.0) with a 685 nm IR-Pass filter. In order to acquire samples with extremely low intensity, the measurement kept one frame, and a neutral-density (ND100) filter with 1% fractional transmittance was added to further reduce the detected photons.

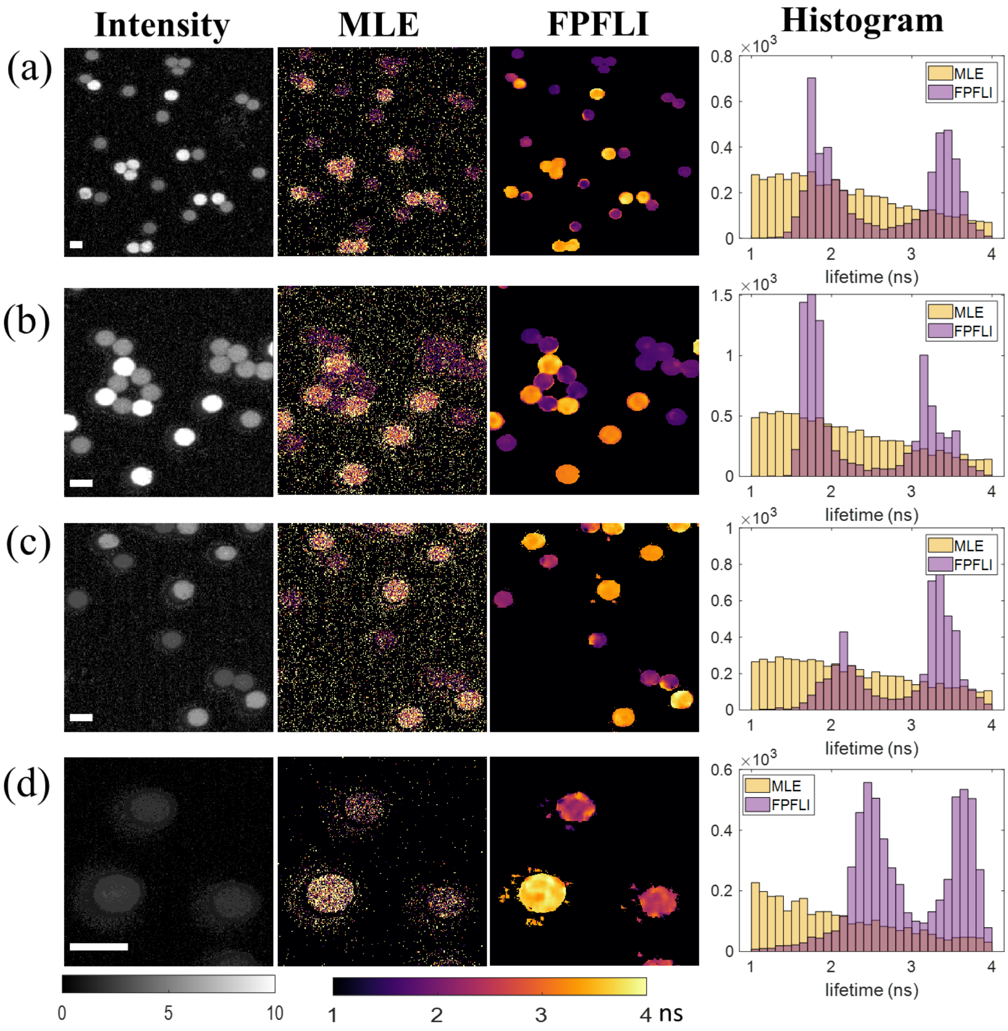


Fig. 5.13. FPFLI for analyzing moving fluorophore-tagged microbeads. (a)-(d) Four FLIM images with different spatial resolutions and average PPPs. The images' size is 256×256. The maximum photon count is 10 for (a) to (c) and 3 for (d). The background noise is significant for all samples and the average PPPs for background noise are 0.4, 0.72, 0.7, 0.32 for (a) to (f). The threshold for MLE analysis is 2.

Fig. 5.13 shows the FLIM samples' intensity images and the corresponding lifetime images calculated by MLE and FPFLI. Due to the different excitation spectra, the crimson and YG beads have different excitation efficiency, leading to a significant intensity difference. The beads can be easily discriminated from the intensity images by eyes. It is tricky to acquire high-intensity FLIM images for the moving beads as GT images. Therefore, we measure the lifetimes of the Crimson and YG microbeads in advance for reference. The Crimson and YG beads have lifetime ranges from 1.9 to 2.1 and from 3.0 to 3.2 ns, respectively. The calculation time for MLE takes around 300 seconds. In contrast, FPFLI only takes less than 2 seconds if the MRF masking is not applied (MRF needs extra 20 seconds). Similar to previous results, MLE cannot analyze the FLIM images as the count is too few and the noise level is too high. By comparison, our FPFLI algorithm can successfully calculate the fluorescence lifetime. The lifetime images and histograms show that FPFLI substantially outperforms MLE and delivers reliable lifetimes. The beads' lifetimes and intensity profiles are well-matched, and the estimated lifetime ranges are consistent with the references. In Fig. 5.13 (d), the photon counts in the images are less than 3, and the noise counts take a large portion. The lifetimes estimated by FPFLI bias towards larger values. Nevertheless, the lifetime images calculated by FPFLI can still provide useful information to discriminate different beads.

5.4.3 Hela cells

The Hela cells transfected with enhanced green fluorescent proteins (EGFPs) only and EGFP fused to mCherry are applied for evaluating our FPFLI algorithm. Here we denote the cell samples as EGFP-HeLa and EGFP-mCherry-HeLa for simplification. HeLa cells were cultured in DMEM and supplemented with 10% heat-inactivated FCS, 100 U/mL penicillin, and 100 μg/mL streptomycin, 1 X Glutamax, and 1 mM sodium pyruvate at 37°C in a 5% CO_2-humidified incubator for 48 hours. Then they were transfected by plasmids encoding EGFP only or an EGFP-mCherry fusion protein after culturing using Turbofect Transfection Reagent (ThermoFisher Scientific) for 24 hours. The cells were fixed in 4% paraformaldehyde and 0.1% glutaraldehyde, and then they were washed in PBS. 50 mM ammonium chloride was used for quenching the autofluorescence. The FLIM images of HeLa cells were acquired by a Leica SP5 scanning confocal microscope with a PicoHarp 300 TCSPC module. The cell samples were excited by a tunable white light laser with a 488 nm wavelength and 40 MHz reputation rate. A dichroic mirror with a 607–683 nm transmitting band and a 500–550 nm reflecting band is used for separating the excitation and emission light. The fluorescence signals were detected by SPAD sensors with a 1.4 NA 63x oil immersion objective lens.

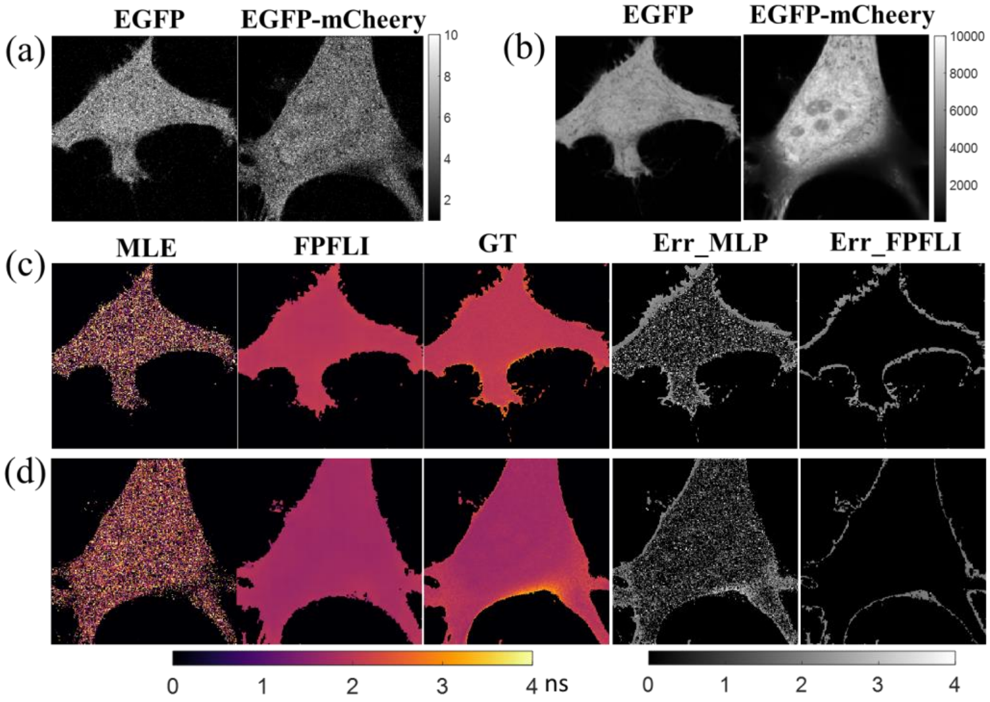


Fig. 5.14. FPFLI for analyzing HeLa cells transfected with EGFP only and EGFP fused to mCherry. (a) and (b) the intensity images of HeLa cells transfected with EGFP only and EGFP fused to mCherry with a low- and high-photon count. The low-photon count images have an average PPP of around 1. The high-photon images are used for calculating GT lifetime images, which have a higher average PPP of around 1600. (c) and (d) the analyzing results of EDFP-HeLa and EGFP-mCherry HeLa for MLE and FPFLI.

Fig. 5.14 shows the lifetime estimation for EGFP-HeLa and EGFP-mCherry-HeLa. Since the cells take up a large portion of the image area and are quite uniform, we use an aggressive binning factor of 16 to combine a larger area for local lifetime analysis. Fig. 5.14 (c) and (d) show the estimated lifetime images using MLE and FPFLI. Undoubtedly, MLE fails again for producing insignificant results. To our surprise, FPFLI's results are almost identical to the GT images. The error map for FPFLI further demonstrates FPFLI's high accuracy. The relatively large errors only come from the boundaries, which have been explained in the previous section. It is shown that even for large binning factors, FPFLI can still work for proper samples.

5.5 Conclusion

In conclusion, this chapter presents and demonstrates a DL algorithm, FPFLI, for fast and robustly analyzing FLIM images in extremely low light conditions. When the photon counts are insufficient, the lifetime estimation becomes an ill-posed problem. It is impossible to find the correct lifetime using conventional FLIM algorithms. We deprecate conventional pixelwise analyzing methods and propose a new lifetime analyzing method using spatial correlation and intensity information. FPFLI algorithm first evaluates the local lifetime information using a dedicated light-weighted DNN. Then it adopts NII to fuse both pieces of information to calculate the lifetime images. The synthetic FLIM images and experimental FLIM images of moving fluorophore-tagged microbeads and HeLa cells are used for evaluating FPFLI's efficient and robust performance. FPFLI paves the way to various photon starve FLIM applications.

Chapter 6

Deep Learning Enhanced Time-resolved Flow Cytometry for Fast Fluorescence Lifetime Sensing

6.1 Introduction

This chapter describes an example application of our proposed DL algorithms, specifically the development of a DL-enhanced time-resolved flow cytometry (TRFC) system for fast fluorescence lifetime sensing. TRFC is an instrument used to measure the fluorescence lifetime of excited fluorophores that pass through the flow chamber, with numerous applications in biomedical sciences, including the study of basic cellular functions, protein conformational changes, and protein-protein interactions. However, this technique has not yet been widely applied due to the low photon-counting throughput and complex data analysis. To address these challenges, this chapter proposes a TRFC system that integrates a CMOS SPAD array and a DL hardware processor. Firstly, the basic concepts of flow cytometry are introduced, and the progress of TRFC is briefly reviewed. Then the whole system design, the development of hardware embedded DL processor, the system calibration, and the performance evaluation are presented in detail.

6.1.1 Flow Cytometry

Flow cytometry is a sophisticated instrument that measures multiple physical parameters of a single cell or particle when it passes through a light source in suspension [182-184]. Over the past 60 years, flow cytometry has evolved to become a billion-dollar market. The earliest flow cytometry was developed for cell counting and was capable of measuring only a single parameter. Modern flow cytometry can simultaneously measure more than a dozen parameters, including both physical and biochemical properties of cells, as well as particles/organelles such as nuclei and microorganisms [185]. Flow cytometry works by exploiting the light scattering properties or fluorescence features of the investigated cell or particles. Light scattering can provide information about the size, shape, and granularity, while fluorescence emission from the fluorescence probes stained on the cells can be used for phenotypic characterization, detection of specified proteins, intracellular cytokine detection, and much more. Flow cytometry can also be applied for fluorescent activated cell sorters (FACS), in which the fluorescent-labeled cells can be sorted from mixed cell steam. Therefore, flow cytometry, as a powerful and versatile tool, has wide applications in various biomedical fields and clinical diagnoses.

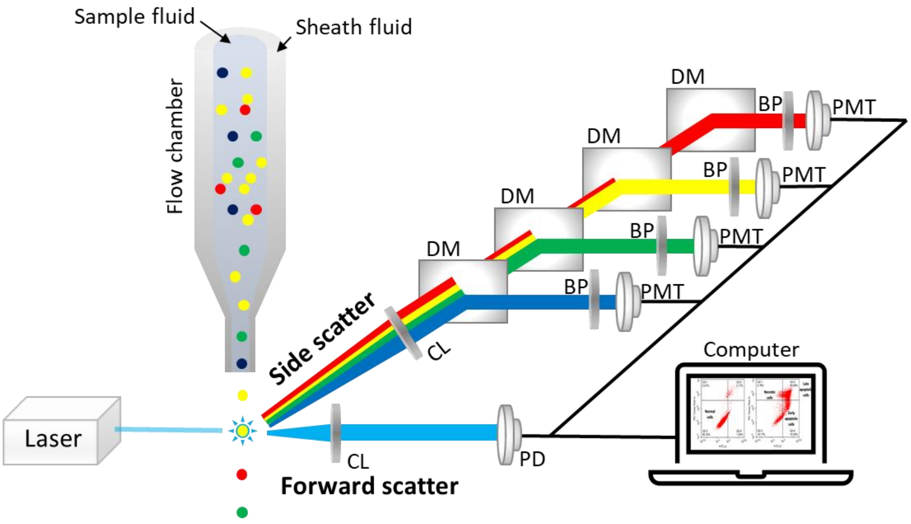


Fig. 6.1. Conceptually schematic of a multi-parameter flow cytometer. Abbreviations for optics components: CL, collecting lens; PD, photodiode; BP, bandpass filter; DM, dichroic mirror; PMT, photomultiplier. Colorful dots represent different cell populations. It should be noted that the DM and BP for different PMTs have different parameters that only allow light with a specific wavelength to pass through.

Fig. 6.1 conceptually shows a typical multi-parameter flow cytometer, which comprised three subsystems: fluid, optical, and electronic. The fluid system is responsible for hydrodynamic focusing on the target cells and delivering them to the light interrogating area. The flow chamber contains two different fluids, the sheath and the sample fluid. Typically, the sheath fluid is PBS diluent that hydrodynamically focuses the sample fluid on the center of the flow chamber. By adjusting the flow rates of the sample and sheath fluid, the pressure of the sample fluid is greater than that of the sheath fluid, forcing the sample fluid into the central core. So, the cells in the sample fluid pass through the laser beam one by one, ensuring uniform illumination of each cell. The diameter of sample fluid d_{sample} can be calculated from [186]:

$$d_{sample} = \sqrt{\frac{f_{sample}}{f_{sheath}}} D_{chamber}, \tag{6.1}$$

where f_{sample} and f_{sheath} represent the sample and sheath flow rate. $D_{chamber}$ is the diameter of the flow chamber.

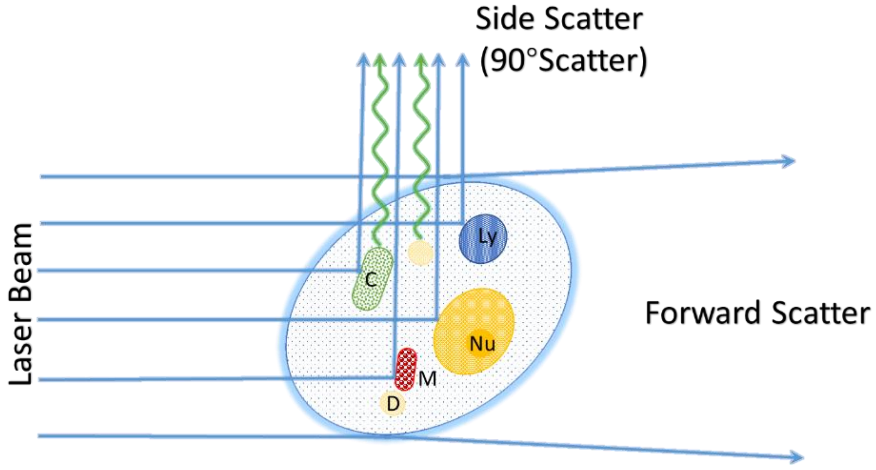


Fig. 6.2. Forward and side scatter of a cell in flow cytometry. Abbreviations for organelles: C, chloroplast; L, lipid body; Ly, lysosome; M, mitochondrion and Nu, nucleolus. The green wavy lines represent the fluorescence or auto-fluorescence.

The optical system comprises a laser source and an optical path that generates and collects scatter or fluorescence light. As shown in Fig. 6.1, in an optical system, the laser source illuminates and excites the cells, while lenses, mirrors, and filters focus, transmit, and select the light. As light hits the cell, two types of scattering light can be observed: forward scatter (FSC) and side scatter (SSC), as shown in Fig. 6.2. FSC is in the same direction as the laser beam and is proportional to the cell size and surface area, providing information for phenotypic characterization. SSC collects the reflected light as well as the fluorescence emission. An optical filter in front of the detector can separate the fluorescence from the scattered light. SSC is proportional to the granularity of the cell, as light can be reflected and refracted by cellular organelles. The fluorescence emission can originate from fluorescent dyes in the membrane, cytoplasmic, or nuclear. or from fluorophore-tagged antibodies. Thus, SSC can be used to analyze pH, membrane potential, enzyme activity, and other properties.

The electronic system comprises photodetectors and data processors, typically computers. For FSC and SSC, two different photodetectors are used. Two different types of photodetectors are used for FSC and SSC: photodiodes with lower sensitivity are used to collect the intense FSC light, while PMTs are used to collect the weaker SSC light. The computer stores the measured data and performs the necessary analysis.

6.1.2 Time-resolved flow cytometer

A time-resolved flow cytometer (TRFC) is a type of flow cytometer that can measure the fluorescence lifetimes of fluorophores as they pass through the light area at a certain speed [187]. Due to its intrinsic properties, fluorescence lifetime as an added cytometry parameter has unparalleled advantages, including:

1. **Increased Quantitative Capabilities**. Fluorescence lifetime is unaffected by the fluorophores' emission intensity, excitation, and emission spectra, making it easier for quantitative measurement. It can also distinguish fluorophores with overlapping emission spectra, providing a route to discriminate different kinds of fluorophore biomarkers bound to cells that cannot be detected by other means like fluorescence intensity-based flow cytometers.
2. **Multifunctional Applications:** As previously discussed, fluorescence lifetime measurement provides a means to probe cellular microenvironments. Thus, TRFC can be used to study various cellular functions, including metabolism, ATP synthesis,

mitochondrial function, and DNA content. It can also be combined with FRET to investigate protein-protein interactions and conformational changes. Fluorescence lifetime could be used as a criterion for cell sorting, enabling the design of novel sorting systems.

Fluorescence lifetime measurement is compatible with other measurements, such as fluorescence intensity and other scatter light, making TRFC a valuable tool for biomedical research and clinical diagnosis due to its versatile functionalities and high throughput. However, accurate lifetime measurement in a flow cytometer poses significant challenges. Flow cytometers require high throughput, and each cell passes through the light area at a high speed and has a limited measurement time. If the collected photons are insufficient, the calculated lifetime may be biased. Thus, there is a trade-off between accuracy and throughput. Lifetime estimation is also a notable concern, and noise from sensors and the environment can pose considerable difficulties in lifetime measurement.

TRFCs can be classified into two types: FD-TRFC and TD-TRFC, based on time-domain and frequency-domain lifetime measurement. FD-TRFC systems have been the mainstream in the past decades because they have few requirements for hardware implementation and data analysis [188]. Earlier FD-TRFC systems were known as phase-sensitive flow cytometry, taking the form of analog homodyning systems. The general principle of FD-TRFC is graphically illustrated in Fig. 6.3 (a). In FD-TRFC, the laser source and fluorescence emission are modulated by a Gaussian function because the cells pass through the continuous laser beam area at a certain speed. The lifetime can be calculated by considering the Gaussian function, fluorescence emission's attenuated intensity, and phase shift. Fig.6.3 (b) shows a general experimental setup of FD-TRFC, where a solid-state laser is used as the light source, and the modulated laser illuminates the cell in the flow chamber through focusing lenses. The scattered light is divided into two branches by a dichroic mirror on the SSC side. The exciting light and fluorescence emissions are separated by optical filters and detected by two independent PMTs. Finally, the electric signals are sent to the data acquisition system for lifetime analysis after pre-amplification.

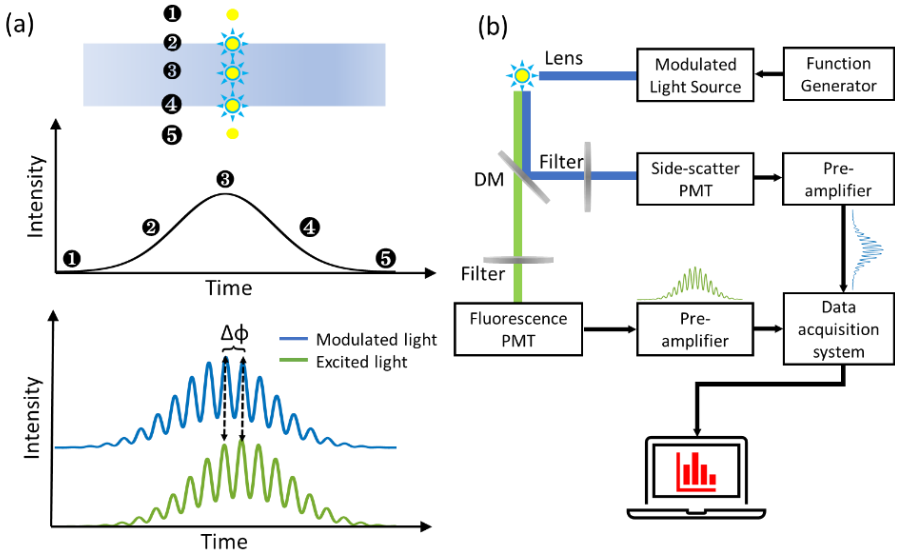


Fig.6.3. (a) Fluorescent particles or cells pass through a continuous laser beam, leading to a Gaussian-shaped intensity profile of the modulated and excited light. The signal intensities depend on the particle's position in the laser beam, as denoted by position ❶ to ❺. (b) The basic components of FD-TRFC systems.

FD-TRFC systems have been extensively used for various applications with different purposes[6]. By measuring the lifetime of organic fluorophores such as propidium iodide, ethidium bromide, and fluorescein isothiocyanate stained on DNA, nucleic acids or antibodies, FD-TRFC can quickly identify and sort specific cells [188-191]. Furthermore, FD-TRFC can measure fluorescent proteins, providing information about protein-protein interactions, protein conformation, gene expressions, and more. For instance, researchers have used FD-TRFC to measure subcellular localization-dependent changes in enhanced green fluorescent protein (EGFP) fluorescence [192]. Additionally, FERT can also apply with FD-TRFC for bioassays. A recent study has reported a high-throughput FERT-based assay for analyzing apoptotic proteases and fluorescence lifetime measurements using FD-TRFC [191].

Recently, TD-TRFC has gained more attention due to the advantages of TD lifetime measurement. In 2013 Li et al. developed a pseudo-time resolved flow cytometry system called the "fluorescence lifetime excitation cytometry by kinetic dithering" (FLECKD) system, which uses an acoustic-optic deflector to dither the continuous laser with high frequency [193].

When a cell passes through the laser focusing area, the dithering laser works like a pulse laser with approximately 25 ns FWHM, allowing the FLECKD system to measure the lifetimes of single exponential fluorescence decays for a series of microspheres and cells. With the advancement of precision timing electronics, ultrafast lasers, and single-photon detectors, some TCSPC-based TD-TRFC systems have been proposed. For example, in 2014, Nebdal et al. applied a microfluidic chip on the FLIM platform and used a picosecond pulsed laser to excite cells [194]. The lifetime of cells was calculated using TCSPC as cells passed through the microfluidic channels. However, the lifetime was post-processed using the nonlinear iterative algorithm. Similarly, in the same year, Léonard et al. also proposed to combine microfluidic chips and the FLIM system [195]. The microfluidic chips were to produce water-in-oil microdroplets containing fluorescein in PBS buffer. The fluorescence was excited by a laser diode with 450 nm wavelength and a 50 MHz repetition rate when the droplets passed through the laser focusing area. The SPAD system detected the fluorescence signal and then post-processed it to extract a lifetime. In 2018, a real-time TCSPC-based TD-TRFC system was reported for lifetime encoding applications that used a modulated laser source and a PMT-based TCSPC system [196]. Unlike previous schemes, the lifetime extraction was a realization by FPGA, providing much faster calculation. However, in this system, the lifetime is represented by the center-of-mass value, which is deviated from the actual lifetime. The PMTs were used as detectors, providing only a single detection channel.

Both FD- and TD-TRFC systems have their advantages and disadvantages. FD-TRFCs are known for their simple instruments and easy data processing, but they suffer from poor temporal resolution and a high photon budget for lifetime evaluation. On the other hand, TD-TRFCs based on TCSPC techniques offer excellent photon efficiency, superior temporal resolution, and high SNR, but they require more sophisticated instruments and complex data processing. Despite some emerging TD-TRFCs reported recently, they are not yet widely used in biomedical research. Integrating TCSPC in TRFCs faces two main challenges:

1. **The photon counting throughput is limited in traditional single-photon detectors (SPDs).** TCSPC suffers from a low photon-counting throughput in traditional SPDs.

Because of SPDs' and time-to-digital converters' dead times, the photon detection rate in standard TCPSC systems should be correctly set (typically below 5% laser repetition rate) to avoid pile-up effects distorting the measured lifetime. Additionally, traditional SPDs, such as PMTs and microchannel plates, have only one or several timing channels, which is not enough for high-throughput fluorescence lifetime measurement. The latest hybrid PMTs with an ultrashort dead-time TDC show a remarkably short dead time that can somewhat improve the photon-counting throughput. But the limited channel number is still inadequate for high-throughput fluorescence lifetime measurement. For the high throughput fluorescence lifetime measurement, cells pass through the laser interrogation point with a high velocity, leading to a limited measurement time. Besides, the illumination is kept low to avoid photo-bleaching. The emitted fluorescence signal, therefore, has a low SNR. Both factors essentially limit the application of TCSPC in TRFCs.

2. **There is a lack of advanced algorithms suitable for real-time and low SNR situations.** Conventional TCPSC systems usually use nonlinear least-square deconvolution or maximum likelihood methods to estimate fluorescence lifetimes from measured fluorescence decays. They are generally computationally intensive and time-consuming, only suitable for offline analysis. Recently, real-time center-of-mass and phasor methods have been widely applied in various applications. However, they are only suitable for high SNR conditions and are sensitive to noise.

6.2 Design of TRFC for fast fluorescence lifetime sensing

6.2.1 System overview

To tackle the two aforementioned challenges, we have designed a TD-TRFC system that employs a CMOS SPAD array sensor and a DL hardware processor. The recent advances in CMOS manufacturing technologies have allowed SPAD array sensors to integrate sensing pixels, timing electronics, and other digital processing units into a single chip. These sensors are compact, operate at low voltage, and consume much less power than traditional SPDs. Furthermore, each pixel in the SPAD array has an independent TDC, allowing for much higher photon-counting throughput since all pixels work in parallel. In our study, we use a 192×128-pixel array CMOS SPAD sensor (QuantiCAM) as a detector for parallel photon detection [62, 197]. Additionally, we have developed a quantized convolution neural network (QCNN) specifically for hardware acceleration, which exhibits high accuracy with a wide dynamic range and low energy/resource consumption. We have also implemented QCNN on FPGA platforms to serve as a DL processor for evaluating fluorescence lifetime.

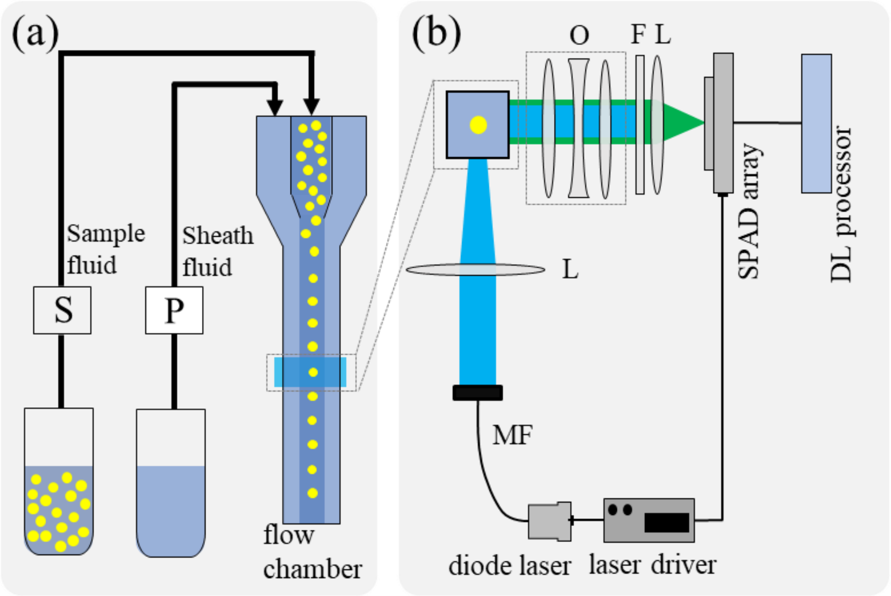


Fig. 6.4. Overview of the TRFC system. (a) The fluidic system. A syringe and a peristaltic pump pumped the sample and sheath flow into the flow chamber, respectively. (b) The optical path and the sensor system. The pulsed diode laser is coupled to multimode fibre (MF) to collimate the light. The laser is focused by an aspheric lens (*L*) and illuminates the flow chamber. The emitted fluorescence signal is collected at the 90° scattered side after passing through an objective lens (*O*), a long-pass filter (*F*), and an aspheric lens.

Fig. 6.4 shows the schematical view of the system, which comprises four subsystems. Fig. 6.4 (a) shows the fluidic system used for delivering the samples to the laser interrogation area sequentially. The optical path in Fig. 6.4 (b) is to collect and pass the fluorescence signals to the detector. The QauntiCAM SPAD system detects and records time-stamped photon events.

The detailed configuration is presented in a later section. An integrated DL processor estimates the fluorescence lifetime.

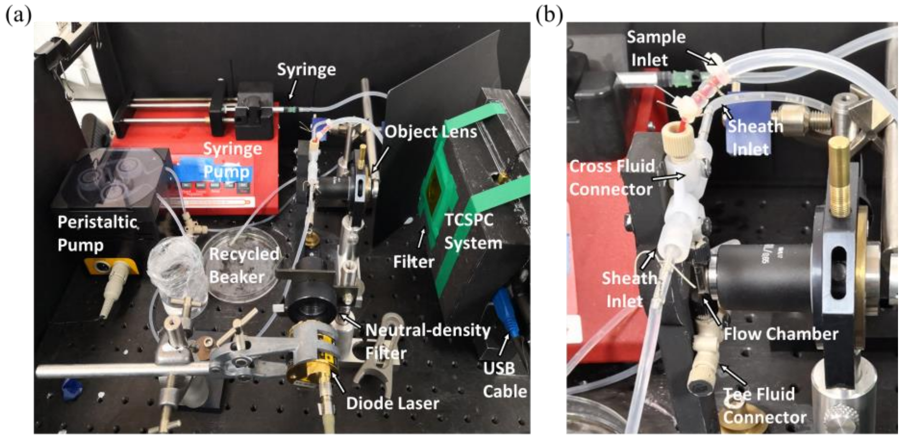


Fig. 6.5. (a) The experimental setup of the TRFC system, (b) partially enlarged view of the flow chamber.

Fig. 6.5 displays the experimental setup of the TRFC system on an optical bench. Each subsystem is composed of discrete components for easier modification. The system is housed in a sealed black box to exclude ambient light and reduce scatter light. For the fluidic subsystem, the sample fluid is injected using a 1 ml syringe with an inner diameter of 4.7 mm (Terumo, UK) controlled by a syringe pump (NE-300, Jaytee, UK). The sheath flow is pumped using a peristaltic pump (Multiflow, Lambda Instrument, Switzerland), driving the sample flow into a central core, and allowing cells/particles to pass the uniformly illuminated area. The flow cytometry cuvette is made of a 250 μm × 250 μm quartz square flow channel (Fireflysci, USA). Fig 6.5 (b) provides an enlarged view of the flowing chamber. The sample and sheath flow is injected simultaneously into the flow cytometry cuvette using a cross-connector assembly (P-634, Kinesis, Dublin, Ireland), and the end of the cuvette connects to the waste line through a tee-connector assembly (P-701, Kinesis, Ireland). The sample and sheath fluid are recycled by a beaker. A MicroLED pulsed diode laser (Horiba Scientific, UK) generated picosecond laser pulses with a pulse width < 200 ps. The fluorescence light is collected at the 90° scattered side by an objective lens (40L/0.65NA, Spindler & Hoyer, Germany) and passes through a long-pass filter. An aspheric lens (Thorlabs) focuses on the fluorescence, and the SPAD sensor is placed on the focal plane to detect fluorescent photons.

6.2.2 *SPAD arrays for high throughput TCSPC*

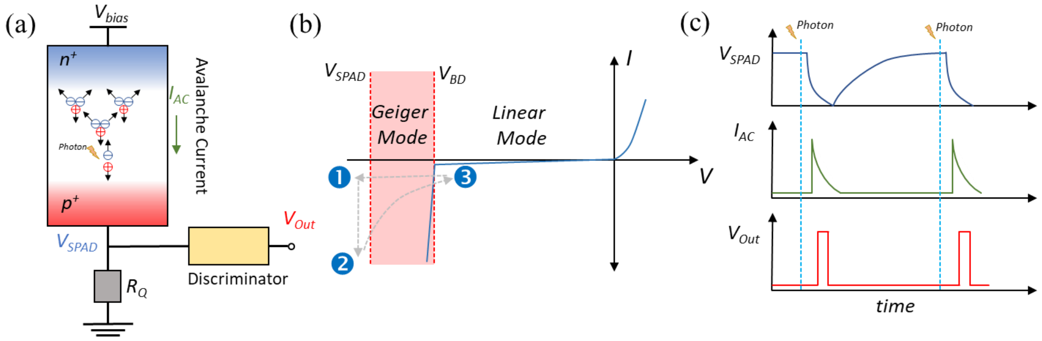


Fig. 6.6. (a) Schematic view of a single photon avalanche diode. Charge carriers undergo multiplication in a SPAD under reverse voltage. (b) Load-line representation of SPAD operation. (c) voltages and currents in (a).

A SPAD is a p-n junction operating in the Geiger-Muller mode, where photoelectrons are generated through the mechanism of avalanche breakdown [198, 199]. Fig. 6.6 (a) shows the schematic drawing of a SPAD. An avalanche diode is simply a p-n junction. The p-type semiconductor has an excess of holes, while the n-type semiconductor has an excess of electrons. This creates a depletion region at the p-n junction where all free carriers are depleted. In the Geiger-Muller mode, the junction is reversed-biased, meaning the n-type semiconductor has a higher potential than the p-type semiconductor. If the voltage across the SPAD (V_{SPAD}) is larger than the breakdown voltage (V_{BD}), the electric field across the depletion region becomes sufficiently high. When a photon is absorbed in the depletion region, drifting electrons are generated and undergo impact ionization, where they gain enough kinetic energy to collide with atoms, generating new electron-hole pairs. Millions of secondary electrons are created during this process, resulting in a self-sustaining avalanche current. The principle of photon detection in a SPAD is shown in Fig. 6.6 (b) [200]. Initially, the SPAD is biased above V_{BD}. When a photon hits the SPAD and the photogenerated charge (electron or hole) triggers the avalanche breakdown, and the avalanche current reaches the peak value within a few picoseconds. The change in voltage associated with this event can be used to precisely measure the photon arriving time. The self-sustaining avalanche should be stopped as soon as possible to prevent damage to the SPAD due to the high current. To reset the SPAD, quenching electronics are used to provide negative feedback and reduce V_{SPAD} below V_{BD}, breaking the condition of the self-sustainable avalanche. A discriminator detects the voltage drop, and an output pulse is generated to record the photon arrival time. Then the SPAD is recharged back to the initial state and can detect the next arriving photon. The corresponding changes in voltages and currents are depicted in Fig. 6.6 (c).

The quenching of a SPAD can be implemented using passive or active methods. Passive quenching involves connecting a high-impedance resistor to the SPAD. The increasing current leads to a significant drop in the voltage, breaking the avalanche when $V_{SPAD} < V_{BD}$. Passive quenching is simple, but it has a very long recovery time in the microsecond range. This recovery time, also known as the dead time, limits the maximum photon counting rate and light intensity. To overcome these drawbacks, active quenching was proposed by adding an active quenching front-end circuit to switch the breakdown voltage. Current active quenching has dead time with a few nanoseconds, which is widely adopted in SPAD sensors and can achieve photon counting rates up to several MHz.

SPADs can be fabricated using custom silicon processes and CMOS techniques. SPADs manufactured by custom silicon processes like custom silicon planar epitaxial technology have achieved a high photon-timing resolution, high photon detection efficiency, and varying area diameters ranging from 5 to 100 μm with a low dark count rate. However, they are large and expensive, making them unsuitable for integrated systems or system-on-chips. Since 2003, SPADs have been fabricated using COMS techniques, making them low-cost, compact, and easy to integrate with other systems. More importantly, they are scalable and can be incorporated into an array to form a SPAD array imager. The SPAD array can be one- and two-dimensional [201]. One- dimensional SPAD array, also known as the SPAD line sensor, is the simplest form of SPAD array. All the detection and processing electronics are placed outside the sensing area, which allows for the highest possible fill factor. However, line sensors require a scanning system to acquire 2D images. Two-dimensional SPAD arrays have more complicated interconnection, readout, and even timing electronics, directly acquiring 2D images. However, these electronics occupy a large portion of the pixel area, resulting in a limited sensing area. Thus, current 2D SPAD arrays suffer from a low fill factor. In addition, they also have relatively large dark-count noise compared to conventional single-photon detectors. SPAD array sensors have been used for various applications, including FLIM, super-resolution microscopy, fluorescence-correlated spectroscopy, time-resolved Raman spectroscopy, and positron emission tomography [201].

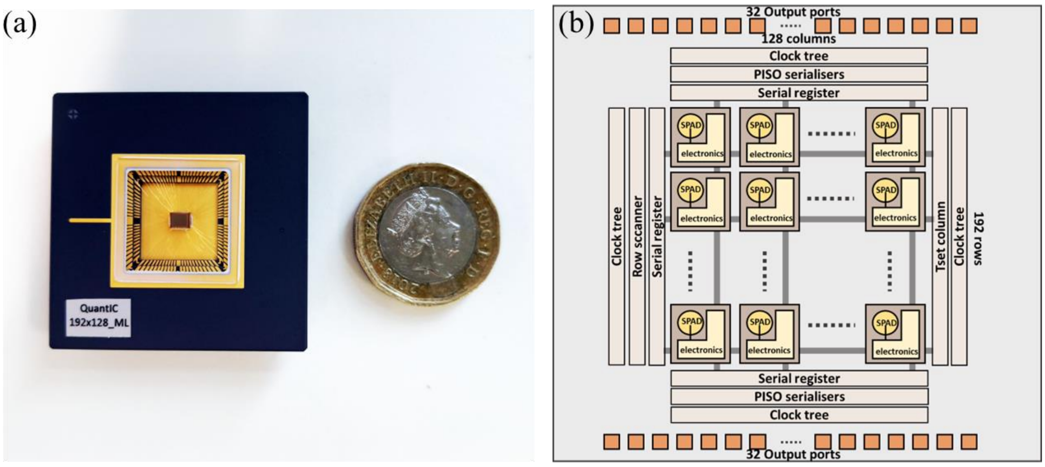


Fig. 6.7. (a) chip layout of the QuantiCAM SPAD sensor. A £1 coin is a reference to indicate its size. (b) Block diagram of the SPAD array in QuantiCAM.

In our study, the 192×128 SPAD array sensor, QuantiCAM, is used for light detection. Prof. Robert K Henderson's research group at the School of Engineering, the University of Edinburgh, provides QuantiCAM and its corresponding firmware and software for configuring the sensor and reading out data. QuantiCAM is fabricated by STMicroelectronics 40-nm CMOS technology. Fig. 6.7 (a) shows the chip layout of the QuantiCAM SPAD sensor. The QuantiCAM sensor comprises a 192×128-pixel array and is integrated into one 3.15 mm × 2.37 mm chip. The chip has two working modes, the photon counting mode and the TCSPC mode. Fig. 6.7 (b) shows the block diagram of the SPAD array in QuantiCAM. Each pixel in the array has an individual 12-bit TDC offering 33 – 120 ps adjustable resolution according to its excess bias. The TDC has a 9.2 μm × 9.2 μm footprint, which is the smallest reported to date. Applying cylindrical microlenses on the chip increases the effective fill factor from 13%

to 42%, thus significantly improving the photon collection efficiency. The chip provides 32×2 output ports at both the top and bottom sides and corresponding addressing circuitry. The detailed design and characterization of this SPAD chip have been reported in [62]. The main parameters of QuantiCAM are listed in Table 6.1.

Table 6.1 Main parameters of QuantiCAM

	Parameter	Value
Chip	Chip size [mm^2]	3.15 × 2.37
	Pixel array	192 × 128
	Array size [μm^2]	18.4 × 9.2
	Maximum frame rate [fps]	18.6k
	Maximum I/O rate [Gbps]	6.4
Pixel	Pixel pitch [μm]	18.4(x), 9.2(y)
	Fill factor/Effective fill factor	13% / 42%
	Peak PDP @ 25°C for 1v bias	34%@560 nm
	TDC depth [bit]	12
	Precision [ps]	321
	Resolution [ps]	33-120
	Median DCR [Hz/V]	25/1.5

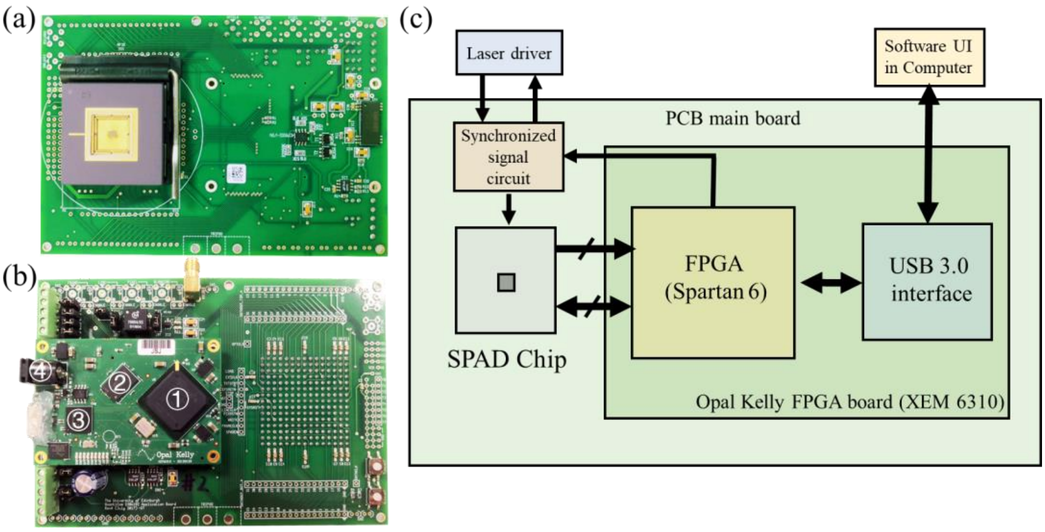


Fig. 6.8. QuantiCAM TCSPC system. (a) The front view. (b) The back view. The daughter board is mounted on the back side of the PCB board. ①-④ denote the Spartan-6 FPGA chip, DDR2 RAM, USB driver, and USB 3.0 interface. (c) The system's block diagram.

Fig. 6.8 shows the QuantiCAM TCSPC system and its block diagram. The QuantiCAM chip is implemented on a PCB mainboard and an FPGA daughterboard (XEM6310, Opal Kelly). The main board is a 6-layered PCB board with a 148 mm × 103 mm footprint. It provides a power supply and an interface for signal communication between the chip and the daughterboard. It also provides an input interface for the laser driver's synchronized signal, which is used as a reference signal for TCSPC timing. Fig. 6.8 (b) shows the FPGA daughter board connected on the backside of the main board by two 80-pin board connectors (BTE-040-01-F-D-A, SAMTEC). The daughter board contains a Xilinx Spartan-6 FPGA, a USB 3.0 interface, 128MB DDR2 memory, and a DC power port to supply the entire TCSPC system. The FPGA provides wide bandwidth for data transfer and sensor configurations. Before the system's operation, the QuantiCAM firmware is downloaded into the FPGA through the USB interface. A MATLAB script configures and communicates with the firmware on a computer. The functionalities of the firmware include configuring the SPAD chip, generating a synchronized signal to lasers, and transmitting and encoding the measured raw data. Fig. 6.9 (c) shows the block diagram of the QuantiCAM TCSPC system. A firmware implemented on the FPGA board is used for 1) generating a synchronization signal to lasers for TCSPC measurement; 2) configuring and controlling the SPAD sensor; 3) collecting, encoding and transmitting the data measured in pixels to the controlling software via USB 3.0 link

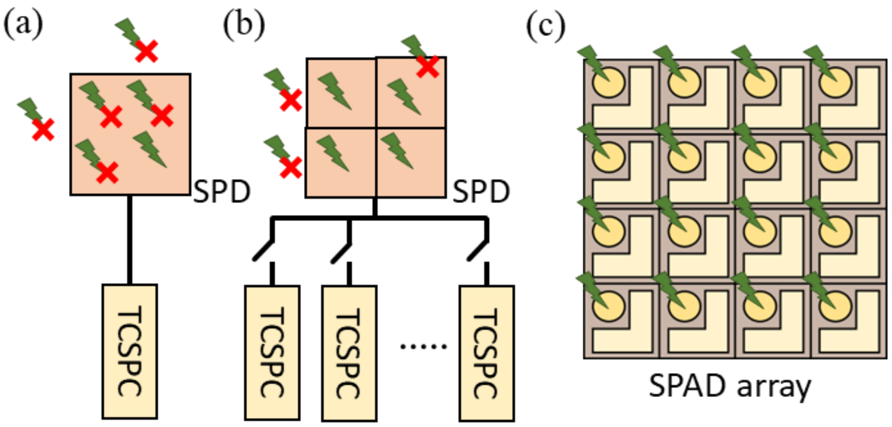


Fig.6.9. Photon counting architectures for TCSPC FLIM. (a) TCSPC with one SPD and one timing channel. The photon counting rate is limited to avoid the pile-up effect. (b)

Multichannel TCSPC with several SPDs and timing channels. (c)Parallel photon counting in SPAD array.

QuantiCAM is initially designed for imaging applications. Here we reconfigure QuantiCAM as a point-like sensor in our TRFC system. The advantages of such a configuration are apparent. Current state-of-the-art CMOS SPAD arrays still suffer from limited sensing areas with low fill factors, high dark count rates, and low photon detection efficiency. QuantiCAM has a large pixel array with a fill factor of 42%, offering better photon collection efficiency. Each pixel has a resolution tunable TDC. All pixels work in parallel, significantly improving the photon throughput. Therefore, it provides a feasible way to address the first challenge mentioned above, the limited photon counting throughput in conventional SPDs. This can be illustrated in Fig. 6.9. As shown in Fig. 6.9 (a), traditional TCSPC-based time-resolved measurement relies on a single SPD like PMT and single-channel TCSPC electronics. The system's long dead time significantly limits the photon counting throughput. Two main sources contribute to the system's overall dead time. The first is the SPD's dead time, typically around tens of nanoseconds. The second is the TCSPC electronics' dead time of several hundred nanoseconds. During the dead time, all arriving photon events will be discarded. If the photon flux is large and multiple photons arrive simultaneously, the measured histogram skews toward a shorter lifetime due to the pile-up effect. So, the instrument should keep a low enough counting rate, which requires a long measurement time to acquire sufficient photons. Such an instrument is unsuitable for high-throughput lifetime measurement.

Numerous efforts have been made to improve the photon counting throughput in TCSPC systems. A direct method is to use a detector and timing electronics with short dead time. For example, the commercially available rapidFLIM system with hybrid PMTs provided by PicoQuant GmbH features very short dead times of less than one ns for both detector and timing electronics, allowing for much higher photon counting rates [202]. Another method is to apply multichannel timing electronics with single or multiple detectors [203, 204]. Fig. 6.9 (b) shows the schematic view of such a configuration. A PMT array is used to detect photons, and the registered photon events are routed to different timing electronics. Thus, the pile-up is largely avoided, and the photon counting throughput is significantly improved. However, the PMT detector array usually has limited detectors, which could be costly and bulky. In addition, there is optical cross-talk between individual detectors in the array, which needs careful calibration. The SPAD arrays revolutionize parallel photon detection, as illustrated in Fig. 6.9 (c). For QuantiCAM, since each pixel has its timing electronics, there are more than twenty thousand timing channels within one chip. Theoretically, if the light uniformly illuminates on the chip, a chip with N pixels can improve the photon counting speed by N-fold compared to single-channel SPDs. However, in practice, the light detection efficiency in SPAD arrays is limited by the low fill factor, high dark count rates, and data transfer rate. Nevertheless, SPAD arrays achieve unprecedented high photon counting rates than traditional TCSPC systems.

6.3 Deep learning processor for fast fluorescence lifetime analysis

6.3.1 *Quantized CNN algorithm*

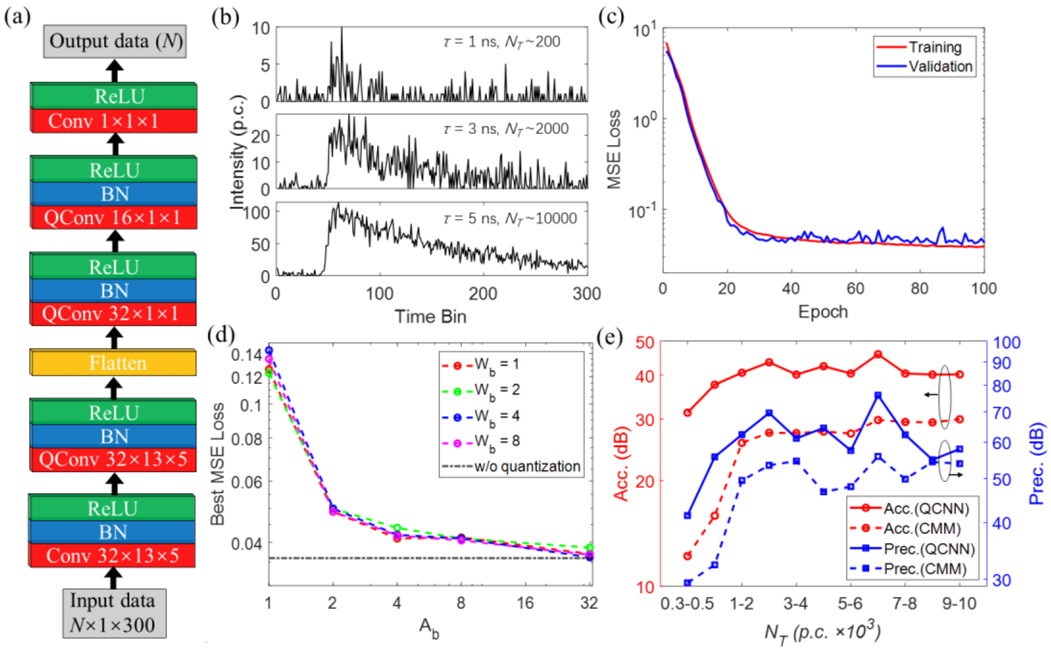


Fig. 6.10. Design and evaluation of the quantized convolutional neural network. (a) The architecture of the neural network. Each block contains a (quantized) convolutional neural network (QConv), a batch normalization (BN) layer, and a rectifier linear unit (ReLU) activation layer. The convolutional layer parameters are the filter number × the kernel size × the stride. (b) Simulated training data with different lifetimes and photon counts. (c) Training and validation of the QCNN with W_b = 1 and A_b = 4. (d) The best MSE loss changes with

different weights and activation quantization bits. (e) Accuracy and precision plots of lifetime determinations using QCNN and CMM.

In chapter 3, we have developed the one-dimensional CNN. This highly efficient network is suitable for lifetime analysis with high throughput, low latency, and low energy consumption. This architecture can rapidly resolve multi-exponential fluorescence decay models with high accuracy. Here we apply it to the TRFC system and further optimize it by quantizing the network parameters for hardware implementation in field programmable gate array devices. The topological structure of the QCNN is shown in Fig. 6.10 (a), which comprises five one-dimensional convolution layers with corresponding BN layers and activation layers (ReLU). Since only average lifetimes need to be measured in our system, the network was tailored for the single-exponential decay model by keeping only one output branch. The first two layers have large kernel sizes and strides to fast capture high-dimensional abstract features of decay functions. The last three layers are pointwise convolution layers for down pooling the information. The three middle convolution layers are quantized to low bit-width weights and activations apart from the first and last convolution layers. Low bit-width quantization is critical for embedded platforms to achieve superior performances in many aspects. Quantizing a neural network can bring enormous benefits while maintaining accuracy. It can increase network throughput by reducing computational complexity and off-chip data transfer. By quantizing the neural network's weights and activations, computationally expensive convolutions are replaced by fast and cost-effective fixed-point arithmetic or bitwise operations, thus significantly improving the calculation speed. Moreover, the parameter size is reduced substantially. It allows the network to use on-chip caching efficiently without relying on slow off-chip memory. The quantization also significantly reduces the consumption of hardware resources and energy.

The weights and activations of layers in the neural network were quantized to an arbitrary bit width to quantise the neural network. Here the "DoReFa" quantization scheme is applied [205]. The k-bit quantization function $Q(r, k)$ for the forward network parameters is expressed as:

$$Q(r,k) = \frac{1}{2^k - 1} roun\,d\left(\left(2^k - 1\right)r\right), \tag{6.2}$$

where ***round()*** is to round the data to the nearest decimal. The input and output of Q are confined in [0, 1]. For the k-bit weight quantization, W_k is expressed as:

$$W_k = 2Q\left(\frac{\tan\mathrm{h}(W)}{2max(|\tanh\,(W)|)} + \frac{1}{2}, k\right) - 1, \tag{6.3}$$

when $k = 1$, W_1 is simplified as $W_1 = \mathrm{sign}(W) \times \boldsymbol{E}(|W|)$. Because the previous batch normalization and ReLU layers ensure the activations are within the range of [0, 1], the k-bit activation is directly obtained by:

$$A_k = Q(A,k). \tag{6.4}$$

Here only the forward propagation is quantized. All the quantized layers follow the "straight-through estimator" method during the training phase for the backwards gradient propagation [206].

A neural network is a data-driven approach. The training dataset determines the resolvable lifetime range and SNR of the QCNN. Besides, obtaining sufficient sample data is also essential for network training. Synthetic fluorescence decays were carefully devised and generated to serve this purpose. In simulations, the number of time bins is $N = 300$ with a bin width $h = 0.04$ ns, similar to the QuantICAM settings. The ground-truth fluorescence decay is expressed as:

$$\mathrm{y(t)} = N_T \cdot IRF(t) * e^{-t/\tau} dt, t = 1, \ldots, N, \tag{6.5}$$

where τ is the fluorescence lifetime ranging from 0.1 to 5 ns with a uniform distribution. The trained lifetime covers a wide range of lifetimes of commonly used fluorophores for biomedical applications. IRF(t) is the equivalent instrument response function of the SPAD array (see details in the next section), approximated by a Gaussian function with FWHM = 230 ps with the peak position at the 50th time bin. The asterisk * denotes the convolution operation and the integral $\int IRF(t) * e^{-t/\tau}\, dt = 1$. Therefore, N_T is the total photon count of the decay, ranging from 100 to 1e4 p.c. The measured decay is the ground-truth decay contaminated by various noise sources. System noise can be generally categorized into Poisson and non-Poisson noise. The former comes from photon detection's discrete nature, whereas the latter has complex origins, including surrounding scattered light, TDC nonlinearity, circuit clock noise, and quantization noise. We used Gaussian noise with zero mean and a standard deviation σ randomly ranging from 1 to $N_T/300$. It is an empirical setting based on observations of decays obtained from the sensor. Thus, we can describe the measured decays as:

$$Y(t) = y(t) + \sqrt{y(t)}\mathcal{N}(0,1) + \mathcal{N}(0,\sigma), \tag{6.6}$$

where $\mathcal{N}(\mu, \sigma)$ is the Gaussian distribution with mean μ and a standard deviation σ. Poisson noise is approximated using a normal distribution. Fig. 6.10 (b) shows synthetic samples with different lifetimes and photon counts. As for the network training, the QCNN was implemented in Python with the open-source library *Pytorch*. The training dataset contains 50,000 samples and 20% of which were for validation. The batch size of the training samples is 128, and the training epochs are 100. The loss function is the MSE, and the optimizer is the Adam algorithm with a learning rate of 1e-4. The MSE losses of training and validation of the QCNN are shown in Fig. 6.10 (c). Here the weight is one-bit, and the activation is four-bit. After 40 epochs, both converged to a plateau with a loss of 0.02. Thanks to the simple and efficient network architecture, the whole training time is less than 3 minutes with the Intel i9 CPU. The effect of the quantization on the network performance is shown in Fig. 6.10 (d). The best MSE loss of the training phase is used as an indicator to evaluate the degradation of the network performance when the parameters are quantized. The bits of weights and activations are denoted as W_b and A_b, respectively. From Fig. 6.10 (d), W_b has little impact on the best MSE loss even if it is quantized to 1 bit. However, a smaller A_b will seriously degrade the network's performance when $A_b < 4$. The results confirm that the one-bit weight and four-bit activation are the optimal configurations for the network design.

A new dataset unseen by the network is used to evaluate QCNN's performance. The sample number, lifetime range, and intensity range are the same as the training dataset. For a better comparison, the lifetimes estimated by QCNN are compared with those calculated by CMM. CMM is the first available hardware algorithm successfully embedded in hardware for fast FLIM analysis, introduced in section 2.4 of Chapter 2. To calculate the lifetimes, CMM starts from the 50th time bin (the peak position) to the 300th time bin. The IRF was considered in lifetime estimations with $\tau_{CMM\_calibrated} = \tau_{CMM} - \tau_{CMM\_IRF}$. To assess the effectiveness of an analysis method, here we defined the accuracy (Acc.) and precision (Prec.) as:

$$Acc.(dB) = 20log_{10}\left(\frac{\tau}{\Delta\tau}\right), \tag{6.7}$$

$$Prec.(dB) = 20log_{10}\left(\frac{\tau}{\sigma\tau}\right), \tag{6.8}$$

where $\Delta\tau$ and $\sigma\tau$ are the absolute error and the standard deviation, respectively. Fig. 6.10 (e) shows the mean accuracy and precision of calculated lifetimes versus N_T. QCNN outperforms CMM in accuracy and precision, especially for low-count conditions ($N_T < 500$). The accuracy of QCNN for N_T from 300 to 500 is even better than that of CMM for N_T from 9000 to 10,000. Additionally, the precision of QCNN also outperforms CMM by a large margin. Although CMM is simple, it is sensitive to noise. It is also nonlinear and can deliver misleading estimations when there are more than two lifetime species in the field of view. Instead, QCNN can circumvent manual settings and directly provide better analysis. It is robust and accurate for noisy and low-count conditions and suitable for various potential applications.

6.3.2 Hardware implementation of the QCNN algorithm

The trained QCNN was implemented on an FPGA as a lifetime processor. FPGAs were first introduced in 1985 by XILINX, Inc. They are integrated circuits whose hardware and function are programmable post-manufacturing by a bitstream. The FPGA is an ideal platform for application-specific integrated circuit (ASIC) prototyping because of its reconfigurability and high-level implementation tools, allowing fast implementation, verification, and upgrade of customized devices. Xilinx and Altera have been the two dominant FPGA providers. FPGA have found their usage in many applications such as wireless communications, video systems, industrial systems, military and aerospace, and big data processing [207].

The basic internal architecture of FPGA is a large array of configurable logic blocks (CLBs) embedded in a programmable interconnect wire network, also referred to as the programmable logic fabric. A CLB consists of one or more programmable logic functions implemented as lookup tables (LUTs), carry chains, and registers. All the components in CLB are configurable by the FPGA's configuration memory to implement the binary functions or sequential circuits. Interconnecting multiple CLBs can realize more complex functionality. The programmable I/O blocks interface the internal signals with the external signals. Contemporary FPGA architectures incorporate additional functional blocks to increase computational density and efficiency, including block memories, digital signal processors (DSPs), high-speed serial transceivers, off-chip memory controllers, Phase-locked loops (PLLs), and so on. Xilinx Zynq FPGA family has integrated a complete system on chip (SoC) as a hard block in the FPGA.

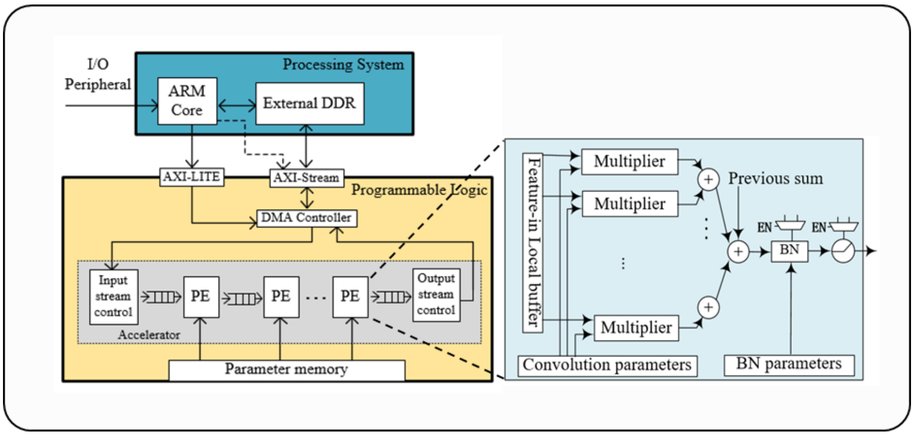


Fig. 6.11. Hardware block diagram for the quantized convolution neural network. The FPGA device can be divided into two parts, the programmable logic (PL) and the processing system (PS), which correspond to the programmable logic circuit and Arm cortex-9 CPU cores. The insert shows the details of the processing element (PE).

This study implements the trained QCNN on an FPGA board as an integrated fluorescence lifetime processor. My collaborator Zhenya Zang developed a DL library using the hardware description language Verilog with Xilinx Vivado High-Level-Synthesis (HLS) tool [208]. The library containing dense, convolution, and BN layers, is adopted to construct the QCNN algorithm in the FPGA board. Fig. 6.11 shows the overall hardware block diagram. The network architecture was designed using the Xilinx Vivado HLS tool. As there are limited digital resources in the Opal Kelly XEM6310 board (with a Xilinx Spartan-6 FPGA), we implemented the DL processor in another FPGA device, ZYNQ 7020 (Xilinx, USA), for this proof-of-concept study. The FPGA device contains programmable logic blocks and dual Arm cortex-9 CPU cores. The former is for establishing the network backbone, whereas the latter is for configuring status registers of the Direct-Memory-Access (DMA) controller and peripherals. The AXI-Stream and AXI-LITE buses are for data transfer and configurations of convolution scale parameters, respectively. Once the model is well trained, all the parameters are parsed and extracted from the model and then initialized in dual-port Block Random-Access Memory (BRAM) modules in the FPGA with processing elements (PE) fetching corresponding parameters for convolution and BN operations. Instead of computing all the input feature data, each PE contains a local buffer that caches a specific length of data to conduct one-time convolution to improve the parallelism. Therefore, all processing elements can execute corresponding operations simultaneously. Additionally, two multiplexers are utilized to configure BN and ReLU modules when they are activated.

Table 6.2 summarizes the main characteristic parameters of the DL lifetime processor. The calculation throughput is up to 54 μs/sample, comparable to a mainstream GPU. As a comparison, the calculation throughput of the Intel i7 4790 is around 55 μs/sample. For a single sample, the total number of floating-point operations (FLOPs) is only 0.2 million, showing the high efficiency of the network architecture. The fast training time allows the processor to be rapidly deployed for different lifetimes and SNR ranges, which is beneficial for broad applications. The DL processor also has compelling advantages in structure simplicity, throughput, and power consumption. It paves the way for portable and compact fluorescence lifetime sensing devices without resorting to slow software tools.

Table 6.2 Main characteristics of DL lifetime

Total parameters	26,881
FLOPs [M/sample]	0.2
Training time [Min]	3
On-chip memory [Mb]	3.37
Throughput [μs/sample]	54

On-chip power [W]	4.5

6.4 Experimental results

6.4.1 system calibration

After setting up the whole TRFC system, the system is calibrated to ensure the correctness of the measurements. The instrument response function (IRF) of the DFLS system was first calibrated. IRF characterizes the system's overall timing precision, which is a function of the uncertainties contributed by the laser excitation, the TCSPC electronics, and the SPD. The IRF is usually characterized by its shape and full width at half maximum (FWHM), typically in several hundred picoseconds. A narrower IRF indicates higher timing resolution. To measure the system's IRF, a solution of Ludox was placed at the laser interrogating point to replace the flow chamber. The scattered light from the Ludox solution with an ultra-short decay was used as the excitation signal. The long-pass filter was removed to allow the scattered light to pass through, and a neutral density filter was added in front of the sensor to reduce the light intensity.

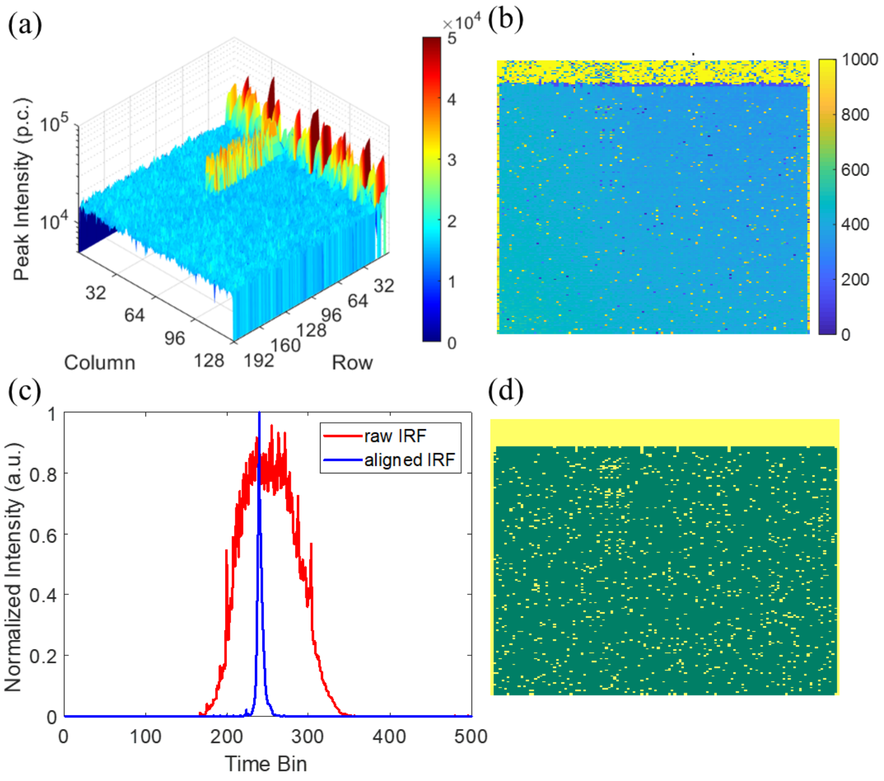


Fig. 6.12. SPAD sensor calibration. (a) The IRF peak intensity distribution. (b) The IRF peak position distribution. (c) The IRF integrated the whole sensor (I-IRF) (in red) and aligned I-IRF (in blue). (d) Mask to exclude hot and silent pixels.

Fig. 6.12 (a) shows the SPAD array's IRF peak intensity distribution; 10^4 frames in total were recorded. Except for the T-shape area that shows abnormal high peak intensities due to manufacturing and readout defects, the sensor array's intensity profile has a relatively flat distribution, indicating that the optical path was well-positioned. Some pixels showing a high dark count rate (usually called hot pixels) exist randomly across the SPAD array. Fig. 6.12 (b) shows the IRF peak positions. The histograms of all pixels are directly truncated from the raw data to observe the peak distribution. Fig. 6.12 (a) and (b) show each pixel and its corresponding TDC behave slightly differently in terms of the response time (related to IRF), noise, jitter, and TDC gain & linearity due to manufacturing uncertainties. Fig. 6.12 (c) shows the integrated IRF (I-IRF) when the sensor array functions as a single megapixel. As indicated by the I-IRF, the sensor array shows an irregular IRF with a significant FWHM over 4 ns. Using such an uncalibrated IRF would severely distort measured decays and lead to incorrect lifetime estimations. To address this problem, pixels must be aligned according to their peak positions. The aligned I-IRF has a sharp peak with an FWHM of 0.23 ns, which is comparable with mainstream FLIM systems. The IRF map also provides helpful information to mask hot and silent pixels, as their IRF profiles (position, FWHM, and background noise level) are significantly different. Fig. 6.12 (d) shows the mask map for the SPAD chip. The intensity peak positions and mask map are recorded in the FPGA for rapid calibrations.

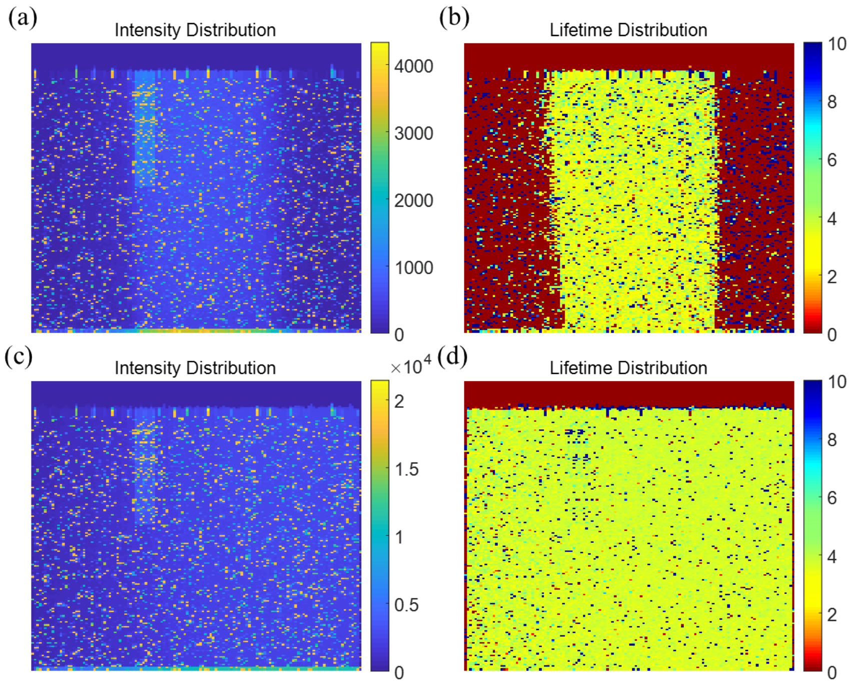


Fig. 6.13. Optical path calibration. The intensity (a) and the lifetime (b) images of the flow chamber before calibration. (c) and (d) the corresponding intensity and lifetime images after calibration.

We further calibrate the optical path to ensure that the sensor can detect the fluorescence signals with the highest efficiency. The aqueous fluorescein solution is pumped into the flow chamber. Fluorescein shows single-exponential decay characteristics with an excellent quantum yield. It is easy to prepare an ideal fluorophore for flow cytometry characterizations. The fluorescein solution was made by mixing fluorescein powder with water until the solution was saturated before injecting it into the flow chamber. Fig.6.13 (a) and (b) show the intensity and lifetime images of the flow chamber. CMM roughly estimates the lifetime without calibration. The lifetime map offers better contrast to observe the chamber image than the intensity image. Although the flow chamber's image is located well in the center of the sensor, it only covers a small portion of the sensor area. The pixels in the chamber margin cannot detect any fluorescence signal. Instead, it will count noise photons from the surrounding environment and the sensor, significantly degrading the performance. The distance between the sensor and chamber should be carefully adjusted to make the chamber image exactly fill in the sensor pixel array to address the problem. Fig. 6.13 (c) and (d) show the flow chamber image after calibrating the optical path.

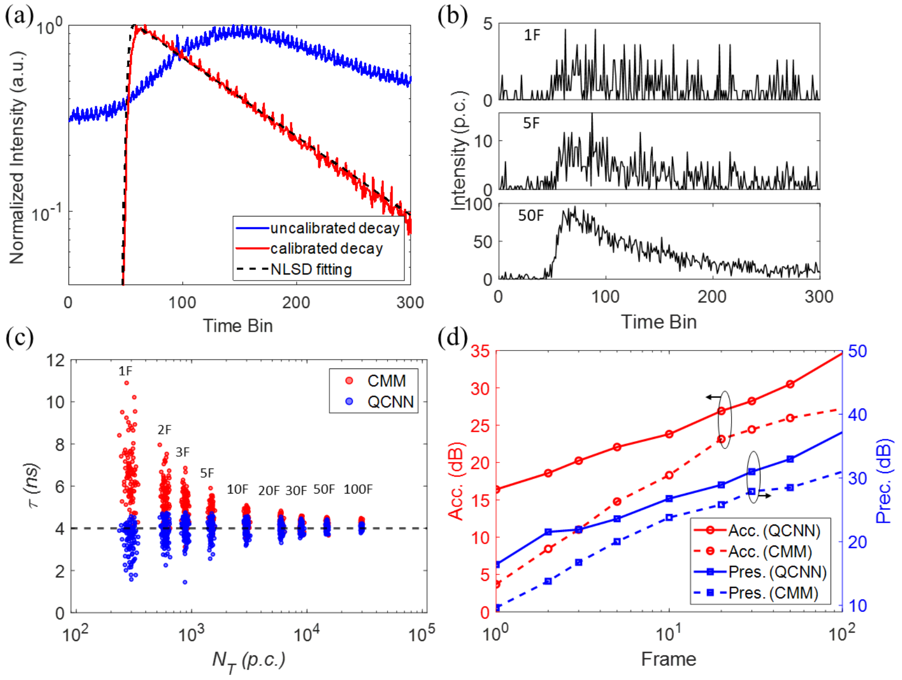


Fig. 6.14. System evaluation. (a) The fluorescence decays of fluorescein before and after calibration. The dashed line is the fitting curve using NLSD. (b) The measured decays under different frames. (c) The lifetime distributions calculated by QCNN and CMM as a function of the total photon count, N_T. The frame number has a linear relationship with N_T. (d) The comparison between QCNN and CMM in terms of accuracy and precision.

The photon-counting throughput and lifetime calculations were also evaluated with aqueous fluorescein solutions. The maximum photon-counting throughput reached 1.9×10^4 counts per frame (cpf) in the experiments. Since we conducted benchmark experiments using fluorophores with known lifetimes, it was easy to ensure that no pile-up effects (fluorescence decay histogram skews towards shorter lifetimes) were triggered. Hence the fluorescence lifetime can be accurately determined within only one frame. An ND filter was inserted to reduce the intensity to around 300 cpf to mimic low-light conditions and investigate the DL processor's performance under low photon counts. Fig. 6.14 (a) shows the fluorescence decay with 10^4 measured frames before and after calibration. The uncalibrated IRF shows considerable background noise and is significantly distorted. As a comparison, after aligning pixel IRFs, removing hot pixels, and removing background noise, the measured decay shows a desirable single-exponential decay. The mean photon count in the first 50 TDC time bins is for estimating background noise. It is worth noting that even though the measured decay is well-calibrated, it still has apparent periodic noise. It is from the TDC nonlinearity and clock noise of sensor electronics. Nevertheless, it has little impact on lifetime estimations. As depicted by the dashed line, the measured decay can be well-fitted by the nonlinear least-squares deconvolution (NLSD) algorithm. The estimated lifetime is 4.01 ns, in good agreement with the literature.

Fig. 6.14 (b) shows the decays under fewer measured frames. Unlike high count conditions, noise from electronics and background scattered light has no periodic distributions. Hence, it can be approximately described as Gaussian noise. The calculated lifetimes of the measured decays are compared using CMM and QCNN. The results are summarized in Figs 6.14 (c) and (d). The fluorescein solution was measured by integrating 1 to 100 frames, and each measurement was repeated 100 times. The measurement window for CMM is from the peak position (50th time bin) to the last time bin. The lifetime versus the total photon count (N_T) is shown in Fig. 6.14 (c). QCNN outperforms CMM significantly under low-count conditions. Interestingly, CMM delivers a more significant value than the ground-truth, whereas QCNN stays closer to the ground-truth. One distinct drawback of CMM is its high sensitivity to noise. It is why, under low-count conditions, CMM delivers more deviated results. On the contrary, QCNN has no such problem and is more robust against complex noise. Fig. 6.14 (d) shows quantitative lifetime analysis in terms of accuracy and precision defined in Eq. (6.7) and (6.8). QCNN delivers better precision and accuracy than CMM, especially for low photon counts.

6.4.2 results and discussion

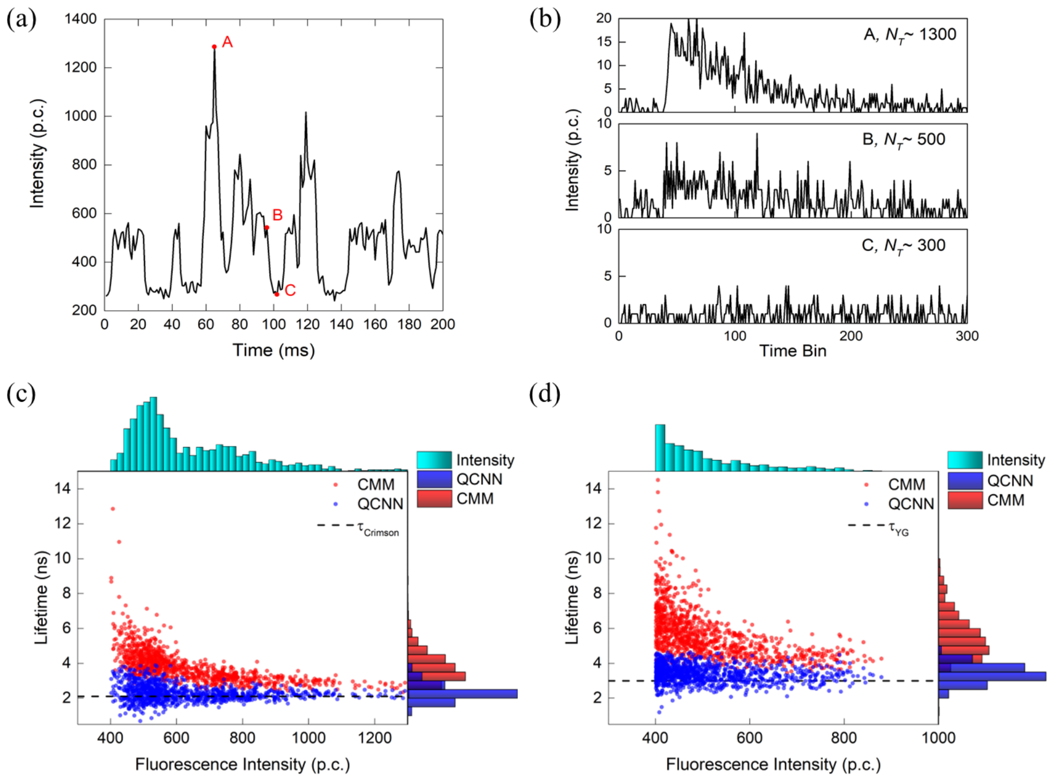


Fig. 6.15. Experimental results of flowing fluorophore-tagged microspheres. (a) The photon detection counts of 200 subsequent frames. Each frame has a 1 ms exposure time. (b) The decay histograms of three different frames denoted in (a). (c) and (d) 2D scatter plots of the fluorescence lifetime versus calculated lifetime using CMM and QCNN for crimson and yellow-green tagged microspheres. The upper- and right-side normalized histograms show fluorescence intensity distributions and lifetimes, respectively. The bin widths of the intensity and lifetime histograms are 20 p.c. and 0.5 ns, respectively.

The developed DFLS system and DL processor were further tested and characterized using fluorophore-tagged microspheres. Two different samples were used in our experiment, one is crimson fluorescent microspheres (F8831, FluoSpheres™ Polystyrene Microspheres, Thermo Fisher, UK), and another is yellow-green (YG) fluorescent microspheres (F8836, FluoSpheres™ Polystyrene Microspheres, Thermo Fisher, UK). Both microspheres have a 10 μm average size and were dissolved in an aqueous solution with a concentration of 3.6×10^5 beads/ml. Before pumping into the flow chamber, the two steady sample solutions' fluorescence lifetimes were measured using a commercial PMT system (FluoroCube Extreme, Horiba Scientific, UK) as a reference. The two samples show a multi-exponential decay feature, and their lifetimes were calculated by commercial software for multiexponential decay fitting (DAS6, Horiba Scientific, UK). The average lifetimes for crimson and YG microspheres were τ_{Crimson} = 2.1 and τ_{YG} = 3.0 ns, respectively. The two sample solutions were separately pumped into the flow chamber at a speed of 0.12 ml/min. For crimson microspheres, the excitation source was a diode laser with a 640 nm peak wavelength, 10 MHz repetition rate and 300mW peak power (DD-635L, Horiba Scientific, UK). A 670 nm long-pass filter was added in front of the SPAD sensor to remove the excitation light. For yellow-green microspheres, the excitation source was changed to a diode laser with a 503 nm peak wavelength, 10 MHz repetition rate and 100mW peak power (DD-510L, Horiba Scientific, UK) and the filter was replaced by a 530 nm long-pass filter.

To measure the fluorescence lifetime of microsphere events in a flow chamber, one principle is the burst integrated fluorescence lifetime (BIFL) analysis method, in which a long exposure time is set, or sequential frames are segmented into different groups to identify events according to a pre-defined fluorescence intensity threshold. However, instead of using BIFL, we analyzed TCSPC data in a single frame with a short exposure time, 1ms per frame, to better evaluate different lifetime estimation algorithms. Fig. 6.15 (a) shows a segment of recorded fluorescence intensity of crimson microspheres for 200 ms. Since a different number of microspheres passed through the laser interrogation area within a frame, the detected fluorescence photons per frame vary significantly from about 300 to 1300 p.c. Figure 6.15 (b) shows fluorescence histograms corresponding to labelled positions *A*, *B*, and *C* on the intensity curve in Fig. 6.15 (a). The measured fluorescence decays are severely contaminated by noise. When no microsphere events are detected, the background noise level is about 300 p.c., mainly contributed by the SPAD dark count. Therefore, an intensity threshold of 400 p.c. is applied to determine detected microsphere events for a frame. It is worth noting that the measured decays of microspheres are similar to the simulated decays shown in Fig. 6.10 (b), indicating that using simulated data is an effective way for network training. Figs 6.15 (c) and (d) show 2D scatter plots for the fluorescence intensity versus calculated lifetime using CMM and QCNN. The upper- and right-sides show normalized histograms for measured fluorescence intensities and lifetimes. The fluorescence intensities of YG microspheres are weaker than those of crimson microspheres. It is due to a lower laser power level for YG microspheres. The scatter plots for both samples show similar distributions. Consistent with previous results, CMM delivers significantly biased results, especially for low photon counts. In contrast, both samples' mean lifetimes calculated by QCNN are close to the reference lifetimes. Table 6.3 summarizes the results calculated by CMM and QCNN. The bias is defined as (τ_{Mean} - τ_{REF})/τ_{REF}, where τ_{Mean} is the mean lifetime calculated by CMM or QCNN and τ_{REF} is the reference lifetime of microspheres. QCNN outperforms CMM in terms of τ_{Mean}, standard deviation (std), and bias by a significant margin. Results show that our DL processor is more suitable for processing fluorescence signals with a low SNR.

Table 6.3 Fluorescence lifetimes of fluorophore-tagged microspheres

fluorophore	τ_{REF} (ns)	CMM			QCNN		
		τ_{Mean} (ns)	*Std*	*Bias*	τ_{Mean}	*Std*	*Bias*
Crimson	2.1	3.82	0.94	81.9%	2.3	0.46	9.52%
Yellow-Green	3.0	5.71	1.59	90.3%	3.4	0.50	13.3%

Experimental results show the advantages of the proposed DFLS systems. Owing to the SPAD array sensor with in-pixel TDCs, an enormous photon-counting throughput is achieved. The fluorescence lifetime can be robustly estimated quickly by sacrificing the investigated cells or particles' spatial information. This is beneficial to applications such as monitoring the dynamic behaviors of a large cell population. In our experiments, the exposure time is 1ms per frame. However, the maximum frame rate of the SPAD array sensor can be 18.6 kframes/s. There is still room to improve the system throughput. The DL processor also shows a great capacity to perform lifetime analysis on low-count and noise-contaminated data. Although only the single-exponential decay was considered in this work, our DL architecture can be extended to multi-exponential decay models by adding more output branches as described in. It can be further applied in FRET analysis in flowing cells or imaging applications. The SPAD camera firmware and the DL processor were implemented in two different FPGA boards for a fast proof-of-concept study. Another noticeable point for the SPAD array sensor is the vast data throughput. As mentioned above, the data transfer rate of our QuantICAM could reach 6.4 Gbps. The data transfer in the current firmware still limits the sensor's performance. Therefore, our DFLS system currently cannot conduct an on-the-fly analysis with low latency. Future developments are required to merge modules accommodating the SPAD array sensor and the DL processor in the same FPGA chip for real-time analysis.

6.5 Conclusion

This chapter provides an introduction to time-resolved flow cytometry and recent developments in the field. It then presents a proof-of-concept TRFC system that employs the QuantiCAM camera and deep learning processors. The system utilizes a 192×128 SPAD array to achieve high TCSPC photon-counting throughput, while the DL processor utilizes quantized convolutional neural networks with simple architecture, low on-chip memory, and power consumption to achieve a high calculation throughput. The DL processor can accurately determine fluorescence lifetimes from low SNR data and outperforms the previously reported hardware algorithm, CMM. The system was calibrated and tested with fluorescein dye solutions, and flowing fluorophore-tagged microspheres were characterized. The TRFC system accurately evaluates fluorescence lifetimes with only one frame and has great potential for high-throughput and real-time applications with further optimization.

Chapter 7

Conclusion and further work

7.1 Summary of thesis achievements

This section summarizes this thesis's key contributions presented through Chapter 3 to Chapter 6. This thesis illustrates that DL brings a data-driven approach to circumventing complex and time-consuming optimization processes, directly mapping the raw data to the targets. This advantage offers us enormous opportunities to revolutionarily change the FLIM analysis paradigm. More importantly, DL techniques can provide novel functionalities and new features that significantly expand FLIM's application range. A summary of this thesis' achievements is presented in the following.

Chapter 3 presents a pixel-wise DL algorithm that uses a high-efficient 1D CNN architecture for fast FLIM analysis. This algorithm achieves state-of-the-art performances, including high accuracy, a fast calculation speed, and a powerful ability to resolve multi-exponential decay models. It can be fast retrained by synthetic data for specific applications. The pixel-wise analysis is more feasible and efficient and can mask unwanted pixels or analyze a selected region of interest. The pixel-wise DL algorithm has a distinct hardware-friendly feature, suitable for edge computing in embedded systems. It opens the door for developing portable, low-cost, and real-time FLIM devices.

A great challenge for developing more advanced FLIM algorithms is a lack of massive high-quality FLIM datasets. Obtaining such datasets is often labor-intensive and tricky. The lifetime dynamic range, lifetime component number, cell morphology, and SNR are restricted due to the limited experimental conditions. Though DL techniques have demonstrated powerful abilities and developed rapidly in recent years, fewer of them have been applied in FLIM applications. Chapter 4 provides a practical solution to address this challenge by developing a computational framework to generate large-scale semi-synthetic FLIM datasets. The key idea is to model TCSPC's physical process and convert fluorescence intensity images to lifetime images. The Human Protein Atlas (HPA) datasets with a wide range of cell types' immunofluorescence intensity images are used for generating FLIM images. The forward TCPSC model can easily control the lifetime range, component number, and SNR. Chapter 4 then presents a DL algorithm, SRI-FLIMnet, which is trained by the semi-synthetic FLIM datasets. SRI-FLIMnet can improve the FLIM image's spatial resolution by mapping the low-resolution input image to its high-resolution counterpart without changing the FLIM system's configuration. Obtaining high-resolution FLIM images usually requires prolonged measurement time. SRI-FLIMnet can alleviate this problem and is promising in high-throughput and real-time FLIM systems.

Chapter 5 further presents an exciting DL algorithm, FPFLI, for robustly estimating FLIM images under unprecedented low-light conditions. Conventionally, a FLIM image requires a high photon budget to calculate the pixel's lifetime accurately. When the photon counts reduce to a few PPPs, it is mathematically impossible to recover the lifetime as the decay information is permanently lost. Therefore, all existing FLIM algorithms are invalid. The extremely low intensity has been considered a prohibited zone for FLIM for a long time. FPFLI adopts a new approach that utilizes the fluorescence lifetime image's spatial correlation and the information in the intensity image to estimate fluorescence lifetime. In FPFLI, we use a hybrid DL technique for different sub-tasks. A redesigned 1D CNN is used for evaluating the local fluorescence lifetime. Then, the neural implicit interpolation technique is used for fusing information of local lifetime and intensity images. We use synthetic and experimental FLIM images to demonstrate FPFLI's superiority in lifetime estimation under a few PPPs conditions.

Chapter 6 presents a proof-of-concept study to illustrate the promising prospects of DL techniques for developing next-generation fluorescence lifetime devices. A prototype of a

TRFC system for fast fluorescence lifetime sensing is designed and investigated. There are two innovations in our design. First, we use a 192×128-pixel array CMOS SPAD sensor (QuantiCAM) as a detector for parallel photon detection, achieving unprecedented high photon counting throughput without a pile-up effect. Second, our TRFC system integrates an embedded DL processor for fast lifetime estimation. The DL processor is designed by implementing the quantized 1D CNN algorithm on an FPGA board, accurately calculating lifetimes from noise-corrupted data. The processor has low power and hardware resource consumption. The TRFC system can find various applications in biomedical research and clinical diagnosis.

7.2 Future work

Although the DL algorithms in this thesis have achieved state-of-the-art performance and encouraging results, there are still some limitations. This section discusses the possible research directions for further improving our study.

- Develop DL algorithms with a wide dynamic range and high accuracy for lifetime estimation.

  In Chapter 3, we show that DL algorithms have unique advantages for fluorescence lifetime estimation. However, the dynamic range of resolvable fluorescence lifetime is limited and determined by a pre-defined range in the training dataset. As shown in Figure 3.6, if the fluorescence lifetime that needs to be estimated is beyond the pre-defined range, the DL algorithm will deliver a wrong result. Meanwhile, one phenomenon observed in Chapter 3 is that the prediction accuracy is inversely proportional to the lifetime dynamic range for a DL algorithm. i.e., a large lifetime dynamic range will lead to worse prediction accuracy and vice versa. There is a tradeoff between the lifetime dynamic range and prediction accuracy. In addition, DL algorithms follow the conventional model-based analysis, largely limiting their application range. For example, a DL algorithm for bi-exponential analysis can only analyze bi-exponential models. It will cause a model-mismatch problem if the decay is single- or multi-exponential. In practical applications, the sample's lifetime components are often unknown, and the number of lifetime components may vary in different pixels. A direct solution to address these problems is to design a DL solver that contains many network parameters trained by datasets with varying lifetime ranges. Because the network's resolvable lifetime range is changeable by loading different parameters without modifying the network's architecture, the lifetime estimation can follow a coarse-fine two-step procedure to ensure a wide dynamic range and high accuracy. The solver can also integrate mono-, bi-, and tri-models for adaptively solving different decay models.

- Develop DL-enhanced spatial resolution algorithms to reconstruct high-frequency contents.

  A significant challenge for the SRI-FLIMnet algorithm in Chapter 4 is reconstructing the high-frequency contents (the fine detail). The fine details in cell images play an important role in many biomedical studies. For LR FLIM images, these details are missing due to the sparse measurement. The SRI-FLIMnet algorithm can well reconstruct the cellular morphology. However, it cannot create finely detailed structures. One possible method to improve the algorithm is to fuse the HR intensity image and LR FLIM image to reconstruct the HR FLIM image. This method is also known as guided super-resolution and is similar to the FPFLI algorithm in Chapter 4. A standard CCD camera can quickly acquire the HR intensity image. Then the intensity image with all the fine details can directly provide information to the network for reconstructing the HR FLIM image. The challenge with this method is that we should modify the FLIM system to obtain intensity and lifetime images simultaneously, which may be costly and complex, especially for laser scanning FLIM systems.

- Exploit local and non-local spatial correlations to develop photon-starved FLIM algorithms.

  In Chapter 5, we demonstrate a powerful DL algorithm, FPFLI, for lifetime reconstruction with a few PPPs. FPFLI uses the local spatial correlation to estimate lifetimes and then fuse the information from the intensity image to reconstruct the FLIM image. A FLIM image possesses not only local spatial correlation but also non-local spatial correlation. The local spatial correlation is apparent as nearby pixels usually have the same lifetime. The non-local spatial correlation indicates that the pixels in the different areas present the same lifetimes, which is also easy to understand. For example, a fluorophore with the same physical state distributed among the whole image tends to have the same lifetime. This property is valuable for robust lifetime estimation, especially in photon-starved conditions. Our further study will take advantage of both the local and non-local spatial correlation to develop more robust algorithms.

- Design compact, low-cost, and real-time FLIM systems

Chapter 6 introduces a prototype design of the TRFC system for fluorescence lifetime sensing as an example. The system currently utilizes two FPGA chips, one for controlling the SPAD sensor (SPAD firmware) and the other for analyzing FLIM data. However, this configuration is expensive and significantly slows down data analysis due to the bottleneck in data transfer between the two FPGA boards. To address these issues, future research should focus on developing more advanced firmware that integrates both functional blocks into a single chip. This integration would eliminate the aforementioned problems and enable the SPAD system with integrated DL processors to perform real-time FLIM measurement and analysis [209]. As a result, the FLIM system would have a smaller footprint and a lower cost, making it suitable for various biomedical research and clinical diagnoses.

Bibliography

[1] B. Amos, "Lessons from the history of light microscopy," *Nat Cell Biol,* vol. 2, no. 8, pp. E151-2, Aug 2000, doi: 10.1038/35019639.

[2] A. J. Wollman, R. Nudd, E. G. Hedlund, and M. C. Leake, "From Animaculum to single molecules: 300 years of the light microscope," *Open Biol,* vol. 5, no. 4, p. 150019, Apr 2015, doi: 10.1098/rsob.150019.

[3] J. W. Lichtman and J. A. Conchello, "Fluorescence microscopy," *Nat Methods,* vol. 2, no. 12, pp. 910-9, Dec 2005, doi: 10.1038/nmeth817.

[4] J. C. Stockert and A. Blazquez-Castro, *Fluorescence Microscopy in Life Sciences*. Bentham Science Publishers, 2017, doi: 10.2174/97816810851801170101.

[5] S. Shashkova and M. C. Leake, "Single-molecule fluorescence microscopy review: shedding new light on old problems," *Biosci Rep,* vol. 37, no. 4, Aug 31 2017, doi: 10.1042/BSR20170031.

[6] T. Terai and T. Nagano, "Fluorescent probes for bioimaging applications," *Curr Opin Chem Biol,* vol. 12, no. 5, pp. 515-21, Oct 2008, doi: 10.1016/j.cbpa.2008.08.007.

[7] J. Zhang, R. E. Campbell, A. Y. Ting, and R. Y. Tsien, "Creating new fluorescent probes for cell biology," *Nat Rev Mol Cell Biol,* vol. 3, no. 12, pp. 906-18, Dec 2002, doi: 10.1038/nrm976.

[8] J. R. Lakowicz, *Principles of fluorescence Spectroscopy*. Springer New York, 2006, doi: 10.1007/978-0-387-46312-4.

[9] H. Wallrabe and A. Periasamy, "Imaging protein molecules using FRET and FLIM microscopy," *Curr Opin Biotechnol,* vol. 16, no. 1, pp. 19-27, Feb 2005, doi: 10.1016/j.copbio.2004.12.002.

[10] D. W. Piston and G. J. Kremers, "Fluorescent protein FRET: the good, the bad and the ugly," *Trends Biochem Sci,* vol. 32, no. 9, pp. 407-14, Sep 2007, doi: 10.1016/j.tibs.2007.08.003.

[11] K. Suhling *et al.*, "Fluorescence lifetime imaging (FLIM): Basic concepts and some recent developments," *Medical Photonics,* vol. 27, pp. 3-40, 2015, doi: 10.1016/j.medpho.2014.12.001.

[12] R. Patalay *et al.*, "Fluorescence lifetime imaging of skin cancer," presented at the Photonic Therapeutics and Diagnostics VII, 2011.

[13] Y. Dancik, A. Favre, C. J. Loy, A. V. Zvyagin, and M. S. Roberts, "Use of multiphoton tomography and fluorescence lifetime imaging to investigate skin pigmentation in vivo," *J Biomed Opt,* vol. 18, no. 2, p. 26022, Feb 2013, doi: 10.1117/1.JBO.18.2.026022.

[14] T. Timm, J. P. von Kries, X. Li, H. Zempel, E. Mandelkow, and E. M. Mandelkow, "Microtubule affinity regulating kinase activity in living neurons was examined by a genetically encoded fluorescence resonance energy transfer/fluorescence lifetime imaging-based biosensor: inhibitors with therapeutic potential," *J Biol Chem,* vol. 286, no. 48, pp. 41711-41722, Dec 2 2011, doi: 10.1074/jbc.M111.257865.

[15] L. Marcu, "Fluorescence lifetime in cardiovascular diagnostics," *J Biomed Opt,* vol. 15, no. 1, p. 011106, Jan-Feb 2010, doi: 10.1117/1.3327279.

[16] L. Sauer, A. S. Vitale, N. K. Modersitzki, and P. S. Bernstein, "Fluorescence lifetime imaging ophthalmoscopy: autofluorescence imaging and beyond," *Eye (Lond),* vol. 35, no. 1, pp. 93-109, Jan 2021, doi: 10.1038/s41433-020-01287-y.

[17] C. Dysli, S. Wolf, M. Y. Berezin, L. Sauer, M. Hammer, and M. S. Zinkernagel, "Fluorescence lifetime imaging ophthalmoscopy," *Prog Retin Eye Res,* vol. 60, pp. 120-143, Sep 2017, doi: 10.1016/j.preteyeres.2017.06.005.

[18] M. Marsden *et al.*, "FLImBrush: dynamic visualization of intraoperative free-hand fiber-based fluorescence lifetime imaging," *Biomed Opt Express,* vol. 11, no. 9, pp. 5166-5180, Sep 1 2020, doi: 10.1364/BOE.398357.

[19] G. J. Bakker, V. Andresen, R. M. Hoffman, and P. Friedl, "Fluorescence lifetime microscopy of tumor cell invasion, drug delivery, and cytotoxicity," *Methods Enzymol,* vol. 504, pp. 109-25, 2012, doi: 10.1016/B978-0-12-391857-4.00005-7.

[20] S. C. Yeh, K. R. Diamond, M. S. Patterson, Z. Nie, J. E. Hayward, and Q. Fang, "Monitoring photosensitizer uptake using two photon fluorescence lifetime imaging microscopy," *Theranostics,* vol. 2, no. 9, pp. 817-26, 2012, doi: 10.7150/thno.4479.

[21] S.-H. Cheng *et al.*, "Recent Advances in Dynamic Monitoring of Drug Release of Nanoparticle Using Förster Resonance Energy Transfer and Fluorescence Lifetime Imaging," *Journal of the Chinese Chemical Society,* vol. 58, no. 6, pp. 798-804, 2011, doi: 10.1002/jccs.201190124.

[22] M. Folmar *et al.*, "Metal enhanced fluorescence of Me-ADOTA·Cl dye by silver triangular nanoprisms on a gold film," *Chemical Physics Letters,* vol. 531, pp. 126-131, 2012, doi: 10.1016/j.cplett.2012.01.087.

[23] R. K. Ahrenkiel, N. Call, S. W. Johnston, and W. K. Metzger, "Comparison of techniques for measuring carrier lifetime in thin-film and multicrystalline photovoltaic materials," *Solar Energy Materials and Solar Cells,* vol. 94, no. 12, pp. 2197-2204, 2010, doi: 10.1016/j.solmat.2010.07.012.

[24] J. H. Huang, F. C. Chien, P. Chen, K. C. Ho, and C. W. Chu, "Monitoring the 3D nanostructures of bulk heterojunction polymer solar cells using confocal lifetime

imaging," *Anal Chem,* vol. 82, no. 5, pp. 1669-73, Mar 1 2010, doi: 10.1021/ac901992c.

[25] J. W. Grate, R. T. Kelly, J. Suter, and N. C. Anheier, "Silicon-on-glass pore network micromodels with oxygen-sensing fluorophore films for chemical imaging and defined spatial structure," *Lab Chip,* vol. 12, no. 22, pp. 4796-801, Nov 21 2012, doi: 10.1039/c2lc40776k.

[26] D. Savateeva, D. Melnikau, V. Lesnyak, N. Gaponik, and Y. P. Rakovich, "Hybrid organic/inorganic semiconductor nanostructures with highly efficient energy transfer," *Journal of Materials Chemistry,* vol. 22, no. 21, 2012, doi: 10.1039/c2jm30917c.

[27] F. Yang, Z. Xu, J. Wang, F. Zan, C. Dong, and J. Ren, "Microwave-assisted aqueous synthesis of new quaternary-alloyed CdSeTeS quantum dots; and their bioapplications in targeted imaging of cancer cells," *Luminescence,* vol. 28, no. 3, pp. 392-400, May-Jun 2013, doi: 10.1002/bio.2395.

[28] K. Guo, S. Achilefu, and M. Y. Berezin, "Dating bloodstains with fluorescence lifetime measurements," *Chemistry,* vol. 18, no. 5, pp. 1303-5, Jan 27 2012, doi: 10.1002/chem.201102935.

[29] D. K. Bird, K. M. Agg, N. W. Barnett, and T. A. Smith, "Time-resolved fluorescence microscopy of gunshot residue: an application to forensic science," *J Microsc,* vol. 226, no. Pt 1, pp. 18-25, Apr 2007, doi: 10.1111/j.1365-2818.2007.01752.x.

[30] A. Ehn, O. Johansson, J. Bood, A. Arvidsson, B. Li, and M. Aldén, "Fluorescence lifetime imaging in a flame," *Proceedings of the Combustion Institute,* vol. 33, no. 1, pp. 807-813, 2011, doi: 10.1016/j.proci.2010.05.083.

[31] D. Comelli, C. D'Andrea, G. Valentini, R. Cubeddu, C. Colombo, and L. Toniolo, "Fluorescence lifetime imaging and spectroscopy as tools for nondestructive analysis of works of art," *Appl Opt,* vol. 43, no. 10, pp. 2175-83, Apr 1 2004, doi:

10.1364/ao.43.002175.

[32] W. Becker, A. Bergmann, and C. Biskup, "Multispectral fluorescence lifetime imaging by TCSPC," *Microsc Res Tech,* vol. 70, no. 5, pp. 403-9, May 2007, doi: 10.1002/jemt.20432.

[33] A. Periasamy, W. Becker, P. T. C. So, A. Bergmann, K. Koenig, and U. Tirlapur, "Picosecond fluorescence lifetime microscopy by TCSPC imaging," presented at the Multiphoton Microscopy in the Biomedical Sciences, 2001.

[34] A. Chessel, F. Waharte, J. Salamero, and C. Kervrann, "A Maximum Likelihood method for lifetime estimation in photon counting-based Fluorescence Lifetime Imaging Microscopy," 21st European Signal Processing Conference (EUSIPCO 2013), Marrakech, Morocco, pp.1-5, 2013.

[35] M. Straume, S. G. Frasier-Cadoret, and M. L. Johnson, "Least-Squares Analysis of Fluorescence Data," in *Topics in Fluorescence Spectroscopy*, (Topics in Fluorescence Spectroscopy, 2002, ch. Chapter 4, pp. 177-240.

[36] M. I. Rowley, A.C.C. Coolen, B. Vojnovic, and P. R. Barber, "Robust Bayesian Fluorescence Lifetime Estimation, Decay Model Selection and Instrument Response Determination for Low-Intensity FLIM Imaging," *PLoS ONE* 11(6): e0158404, 2016, doi.org/10.1371/journal.pone.0158404

[37] M. A. Digman, V. R. Caiolfa, M. Zamai, and E. Gratton, "The phasor approach to fluorescence lifetime imaging analysis," *Biophys J,* vol. 94, no. 2, pp. L14-6, Jan 15 2008, doi: 10.1529/biophysj.107.120154.

[38] C. Stringari, A. Cinquin, O. Cinquin, M. A. Digman, P. J. Donovan, and E. Gratton, "Phasor approach to fluorescence lifetime microscopy distinguishes different metabolic states of germ cells in a live tissue," *Proc Natl Acad Sci U S A,* vol. 108, no. 33, pp. 13582-7, Aug 16 2011, doi: 10.1073/pnas.1108161108.

[39] D. D. Li et al., "Video-rate fluorescence lifetime imaging camera with CMOS single-photon avalanche diode arrays and high-speed imaging algorithm," *J Biomed Opt*, vol. 16, no. 9, p. 096012, Sep 2011, doi: 10.1117/1.3625288.

[40] S. P. Poland et al., "New high-speed centre of mass method incorporating background subtraction for accurate determination of fluorescence lifetime," *Opt Express*, vol. 24, no. 7, pp. 6899-915, Apr 4 2016, doi: 10.1364/OE.24.006899.

[41] G. G. Stokes, "On the change of refrangibility of light," Philos. Trans. R. Soc. Lond. 142, pp.463-562. 1852.

[42] Sytsma, Vroom, G. De, and Gerritsen, "Time-gated fluorescence lifetime imaging and microvolume spectroscopy using two-photon excitation," *Journal of Microscopy,* vol. 191, no. 1, pp. 39-51, 2008, doi: 10.1046/j.1365-2818.1998.00351.x.

[43] W. Becker, *Advanced Time-Correlated Single Photon Counting Techniques*. 2015.

[44] P. Kapusta, M. Wahl, and R. Erdmann, *Advanced Photon Counting* (Springer Series on Fluorescence). 2015.

[45] W. Becker, *The bh TCSPC handbook. 9th edition available on www.becker-hickl.com*. 2021.

[46] J. Arlt, D. Tyndall, B. R. Rae, D. D. Li, J. A. Richardson, and R. K. Henderson, "A study of pile-up in integrated time-correlated single photon counting systems," *Rev Sci Instrum,* vol. 84, no. 10, p. 103105, Oct 2013, doi: 10.1063/1.4824196.

[47] M. Kress *et al.*, "Time-resolved microspectrofluorometry and fluorescence lifetime imaging of photosensitizers using picosecond pulsed diode lasers in laser scanning microscopes," *J Biomed Opt,* vol. 8, no. 1, pp. 26-32, Jan 2003, doi: 10.1117/1.1528595.

[48] R. Datta, T. M. Heaster, J. T. Sharick, A. A. Gillette, and M. C. Skala, "Fluorescence lifetime imaging microscopy: fundamentals and advances in instrumentation, analysis,

and applications," *J Biomed Opt,* vol. 25, no. 7, pp. 1-43, May 2020, doi: 10.1117/1.JBO.25.7.071203.

[49] K. Suhling *et al.*, "Wide-field time-correlated single photon counting-based fluorescence lifetime imaging microscopy," *Nucl Instrum Methods Phys Res A,* vol. 942, p. 162365, Oct 21 2019, doi: 10.1016/j.nima.2019.162365.

[50] N. Oleksiievets *et al.*, "Wide-Field Fluorescence Lifetime Imaging of Single Molecules," *J Phys Chem A,* vol. 124, no. 17, pp. 3494-3500, Apr 30 2020, doi: 10.1021/acs.jpca.0c01513.

[51] J. B. Pawley, *Handbook Of Biological Confocal Microscopy*. 2006.

[52] A. Nwaneshiudu, C. Kuschal, F. H. Sakamoto, R. R. Anderson, K. Schwarzenberger, and R. C. Young, "Introduction to confocal microscopy," *J Invest Dermatol,* vol. 132, no. 12, p. e3, Dec 2012, doi: 10.1038/jid.2012.429.

[53] U. Tauer, "Advantages and risks of multiphoton microscopy in physiology," *Exp Physiol,* vol. 87, no. 6, pp. 709-14, Nov 2002, doi: 10.1113/eph8702464.

[54] A. Ulku *et al.*, "Wide-field time-gated SPAD imager for phasor-based FLIM applications," *Methods Appl Fluoresc,* vol. 8, no. 2, p. 024002, Feb 5 2020, doi: 10.1088/2050-6120/ab6ed7.

[55] J. Siegel *et al.*, "Wide-field time-resolved fluorescence anisotropy imaging (TR-FAIM): Imaging the rotational mobility of a fluorophore," *Review of Scientific Instruments,* vol. 74, no. 1, pp. 182-192, 2003, doi: 10.1063/1.1519934.

[56] L. M. Hirvonen and K. Suhling, "Wide-field TCSPC: methods and applications," *Measurement Science and Technology,* vol. 28, no. 1, 2017, doi: 10.1088/1361-6501/28/1/012003.

[57] M. Caccia, L. Nardo, R. Santoro, and D. Schaffhauser, "Silicon Photomultipliers and SPAD imagers in biophotonics: Advances and perspectives," *Nuclear Instruments and*

Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment, vol. 926, pp. 101-117, 2019, doi: 10.1016/j.nima.2018.10.204.

[58] C. Bruschini, H. Homulle, I. M. Antolovic, S. Burri, and E. Charbon, "Single-photon avalanche diode imagers in biophotonics: review and outlook," *Light Sci Appl,* vol. 8, p. 87, 2019, doi: 10.1038/s41377-019-0191-5.

[59] Parmesan Luca *et al.*, "A 256 x 256 SPAD array with in-pixel Time to Amplitude Conversion for Fluorescence Lifetime Imaging Microscopy," presented at the International Image Sensor Workshop Vaals, Netherlands, 2015.

[60] D. E. Schwartz, E. Charbon, and K. L. Shepard, "A Single-Photon Avalanche Diode Array for Fluorescence Lifetime Imaging Microscopy," *IEEE J Solid-State Circuits,* vol. 43, no. 11, pp. 2546-2557, Nov 21 2008, doi: 10.1109/JSSC.2008.2005818.

[61] D. Stoppa, D. Mosconi, L. Pancheri, and L. Gonzo, "Single-Photon Avalanche Diode CMOS Sensor for Time-Resolved Fluorescence Measurements," *IEEE Sensors Journal,* vol. 9, no. 9, pp. 1084-1090, 2009, doi: 10.1109/jsen.2009.2025581.

[62] R. K. Henderson *et al.*, "A 192 x 128 Time Correlated SPAD Image Sensor in 40-nm CMOS Technology," *IEEE Journal of Solid-State Circuits,* vol. 54, no. 7, pp. 1907-1916, 2019, doi: 10.1109/jssc.2019.2905163.

[63] J. Richardson *et al.*, "A 32 x 32 50ps resolution 10 bit time to digital converter array in 130nm CMOS for time correlated imaging," presented at the 2009 IEEE Custom Integrated Circuits Conference, 2009.

[64] M. Straume, S. G. Frasier-Cadoret, and M. L. Johnson, "Least-Squares Analysis of Fluorescence Data," in *Topics in Fluorescence Spectroscopy*, (Topics in Fluorescence Spectroscopy, 2002, ch. Chapter 4, pp. 177-240).

[65] K. Santra, J. Zhan, X. Song, E. A. Smith, N. Vaswani, and J. W. Petrich, "What Is the Best Method to Fit Time-Resolved Data? A Comparison of the Residual Minimization

and the Maximum Likelihood Techniques As Applied to Experimental Time-Correlated, Single-Photon Counting Data," *The Journal of Physical Chemistry B,* vol. 120, no. 9, pp. 2484-2490, 2016, doi: 10.1021/acs.jpcb.6b00154.

[66] J. A. Jo, Q. Fang, and L. Marcu, "Ultrafast Method for the Analysis of Fluorescence Lifetime Imaging Microscopy Data Based on the Laguerre Expansion Technique," *IEEE J Quantum Electron,* vol. 11, no. 4, pp. 835-845, 2005, doi: 10.1109/JSTQE.2005.857685.

[67] Y. Zhang, Y. Chen, and D. D. Li, "Optimizing Laguerre expansion based deconvolution methods for analysing bi-exponential fluorescence lifetime images," *Opt Express,* vol. 24, no. 13, pp. 13894-905, Jun 27 2016, doi: 10.1364/OE.24.013894.

[68] A. S. Dabir, C. A. Trivedi, Y. Ryu, P. Pande, and J. A. Jo, "Fully automated deconvolution method for on-line analysis of time-resolved fluorescence spectroscopy data based on an iterative Laguerre expansion technique," *J Biomed Opt,* vol. 14, no. 2, p. 024030, Mar-Apr 2009, doi: 10.1117/1.3103342.

[69] E. Bajzer, T. M. Therneau, J. C. Sharp, and F. G. Prendergast, "Maximum likelihood method for the analysis of time-resolved fluorescence decay curves," *European Biophysics Journal,* vol. 20, no. 5, pp. 247-262, 1991, doi: 10.1007/bf00450560.

[70] T. A. Laurence and B. A. Chromy, "Efficient maximum likelihood estimator fitting of histograms," *Nat Methods,* vol. 7, no. 5, pp. 338-9, May 2010, doi: 10.1038/nmeth0510-338.

[71] A. Periasamy *et al.*, "Bayesian analysis of fluorescence lifetime imaging data," presented at the Multiphoton Microscopy in the Biomedical Sciences XI, 2011.

[72] B. Kaye, P. J. Foster, T. Y. Yoo, and D. J. Needleman, "Developing and Testing a Bayesian Analysis of Fluorescence Lifetime Measurements," *PLoS One,* vol. 12, no. 1, p. e0169337, 2017, doi: 10.1371/journal.pone.0169337.

[73] R. M. Ballew and J. N. Demas, "An error analysis of the rapid lifetime determination method for the evaluation of single exponential decays," *Analytical Chemistry,* vol. 61, no. 1, pp. 30-33, 2002, doi: 10.1021/ac00176a007.

[74] S. P. Chan, Z. J. Fuller, J. N. Demas, and B. A. DeGraff, "Optimized gating scheme for rapid lifetime determinations of single-exponential luminescence lifetimes," *Anal Chem,* vol. 73, no. 18, pp. 4486-90, Sep 15 2001, doi: 10.1021/ac0102361.

[75] D. D. Li *et al.*, "Time-domain fluorescence lifetime imaging techniques suitable for solid-state imaging sensor arrays," *Sensors (Basel),* vol. 12, no. 5, pp. 5650-69, 2012, doi: 10.3390/s120505650.

[76] S. F. Silva, J. P. Domingues, and A. M. Morgado, "Can we use rapid lifetime determination for fast, fluorescence lifetime based, metabolic imaging? Precision and accuracy of double-exponential decay measurements with low total counts," *PLoS One,* vol. 14, no. 5, p. e0216894, 2019, doi: 10.1371/journal.pone.0216894.

[77] D. D. Li, B. R. Rae, R. Andrews, J. Arlt, R. K. Henderson, "Hardware implementation algorithm and error analysis of high-speed fluorescence lifetime sensing systems using center-of-mass method," *J Biomed Opt* 15(1) 017006, January 2010, doi.org/10.1117/1.3309737

[78] D. D. Li, H. Yu, and Y. Chen, "Fast bi-exponential fluorescence lifetime imaging analysis methods," *Opt Lett,* vol. 40, no. 3, pp. 336-9, Feb 1 2015, doi: 10.1364/OL.40.000336.

[79] D. U. Li, E. Bonnist, D. Renshaw, and R. Henderson, "On-chip, time-correlated, fluorescence lifetime extraction algorithms and error analysis," *J Opt Soc Am A Opt Image Sci Vis,* vol. 25, no. 5, pp. 1190-8, May 2008, doi: 10.1364/josaa.25.001190.

[80] D. U. Li *et al.*, "Hardware implementation and calibration of background noise for an integration-based fluorescence lifetime sensing algorithm," *J Opt Soc Am A Opt Image*

Sci Vis, vol. 26, no. 4, pp. 804-14, Apr 2009, doi: 10.1364/josaa.26.000804.

[81] S. Legg and M. Hutter, "A Collection of Definitions of Intelligence," *arXiv:0706.3639,* 2007.

[82] A. Hodges, "Alan Turing and the Turing Test," in *Parsing the Turing Test*, ch. 2, pp. 13-22, 2009.

[83] J. McCarthy, M. L. Minsky, N. Rochester, and C. E. Shannon. (2006) "A Proposal for the Dartmouth Summer Research Project on Artificial Intelligence, August 31, 1955," *AI MAGAZINE*. 27, pp. 12-14, 2006, doi:10.1609/aimag.v27i4.1904.

[84] M. I. Jordan and T. M. Mitchell, "Machine learning: Trends, perspectives, and prospects," *Science,* vol. 349, no. 6245, pp. 255-260, 2015, doi: 10.1126/science.aaa841.

[85] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature,* vol. 521, no. 7553, pp. 436-44, May 28 2015, doi: 10.1038/nature14539.

[86] M. Minsky, "Steps toward Artificial Intelligence," in Proceedings of the IRE, vol. 49, no. 1, pp. 8-30, Jan. 1961, doi: 10.1109/JRPROC.1961.287775.

[87] "https://www.tensorflow.org/." (accessed 2022).

[88] "https://pytorch.org/." (accessed 2022).

[89] "https://mxnet.apache.org/versions/1.9.1/." (accessed 2022).

[90] "https://www.paddlepaddle.org.cn/." (accessed 2022).

[91] F. A. Azevedo *et al.*, "Equal numbers of neuronal and nonneuronal cells make the human brain an isometrically scaled-up primate brain," *J Comp Neurol,* vol. 513, no. 5, pp. 532-41, Apr 10 2009, doi: 10.1002/cne.21974.

[92] X. Glorot, A. Bordes, and Y. Bengio, "Deep Sparse Rectifier Neural Networks," in

Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics, PMLR, 2011, vol. 15, pp. 315-323.

[93] A. L. Maas, A. Y. Hannun;, and A. Y. Ng, "Rectifier nonlinearities improve neural network acoustic models," in *Proc. icml.* , 2013, vol. 30.

[94] D.-A. Clevert, T. Unterthiner, and S. Hochreiter, "Fast and Accurate Deep Network Learning by Exponential Linear Units (ELUs)," *arXiv:1511.07289* 2015.

[95] P. Ramachandran, B. Zoph, and Q. V. Le, "Swish: a Self-Gated Activation Function," *arXiv:1710.05941,* 2017.

[96] D. Hendrycks and K. Gimpel, "Gaussian Error Linear Units (GELUs)," *arXiv:1606.08415,* 2020.

[97] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016.

[98] K. P. Murphy, *Probabilistic Machine Learning: An Introduction*. MIT Press, 2022.

[99] G. Cybenko, "Approximation by superpositions of a sigmoidal function," *Mathematics of Control, Signals, and Systems,* vol. 2, no. 4, pp. 303-314, 1989, doi: 10.1007/bf02551274.

[100] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," *arXiv:1409.1556,* 2014.

[101] D. Bank, N. Koenigstein, and R. Giryes, "Autoencoders," *arXiv:2003.05991,* 2021.

[102] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional Networks for Biomedical Image Segmentation," *arXiv:1505.04597,* 2015.

[103] I. J. Goodfellow *et al.*, "Generative Adversarial Networks," *arXiv:1406.2661,* 2014.

[104] A. Dash, J. Ye, and G. Wang, "A review of Generative Adversarial Networks (GANs) and its applications in a wide variety of disciplines -- From Medical to Remote

Sensing," *arXiv:2110.01442,* 2021.

[105] A. Vaswani *et al.*, "Attention Is All You Need," *arXiv:1706.03762,* 2017.

[106] Y. W. Tianyang Lin, Xiangyang Liu, Xipeng Qiu, "A Survey of Transformers," *arXiv:2106.04554,* 2021.

[107] R. Sun, "Optimization for deep learning: theory and algorithms," *arXiv:1912.08957,* 2019.

[108] M. Weigert *et al.*, "Content-aware image restoration: pushing the limits of fluorescence microscopy," *Nat Methods,* vol. 15, no. 12, pp. 1090-1097, Dec 2018, doi: 10.1038/s41592-018-0216-7.

[109] Y. Xue, S. Cheng, Y. Li, and L. Tian, "Reliable deep-learning-based phase imaging with uncertainty quantification," *Optica,* vol. 6, no. 5, pp. 618-619, May 2019, doi: 10.1364/optica.6.000618.

[110] Y. Wu *et al.*, "Three-dimensional virtual refocusing of fluorescence microscopy images using deep learning," *Nat Methods,* vol. 16, no. 12, pp. 1323-1331, Dec 2019, doi: 10.1038/s41592-019-0622-5.

[111] L. Jin *et al.*, "Deep learning enables structured illumination microscopy with low light levels and enhanced speed," *Nat Commun,* vol. 11, no. 1, p. 1934, Apr 22 2020, doi: 10.1038/s41467-020-15784-x.

[112] S. Lee, M. Negishi, H. Urakubo, H. Kasai, and S. Ishii, "Mu-net: Multi-scale U-net for two-photon microscopy image denoising and restoration," *Neural Netw,* vol. 125, pp. 92-103, May 2020, doi: 10.1016/j.neunet.2020.01.026.

[113] Y. Rivenson, Z. Göröcs, H. Günaydin, Y. Zhang, H. Wang, and A. Ozcan, "Deep learning microscopy," *Optica,* vol. 4, no. 11, 2017, doi: 10.1364/optica.4.001437.

[114] H. Wang *et al.*, "Deep learning enables cross-modality super-resolution in

fluorescence microscopy," *Nat Methods,* vol. 16, no. 1, pp. 103-110, Jan 2019, doi: 10.1038/s41592-018-0239-0.

[115] L. Heinrich, J. A. Bogovic, and S. Saalfeld, "Deep Learning for Isotropic Super-Resolution from Non-isotropic 3D Electron Microscopy," presented at the Medical Image Computing and Computer-Assisted Intervention − MICCAI 2017: 20th International Conference, Quebec City, QC, Canada, 2017.

[116] H. Byeon, T. Go, and S. J. Lee, "Deep learning-based digital in-line holographic microscopy for high resolution with extended field of view," *Optics & Laser Technology,* vol. 113, pp. 77-86, 2019, doi: 10.1016/j.optlastec.2018.12.014.

[117] E. Nehme, L. E. Weiss, T. Michaeli, and Y. Shechtman, "Deep-STORM: super-resolution single-molecule microscopy by deep learning," *Optica,* vol. 5, no. 4, 2018, doi: 10.1364/optica.5.000458.

[118] W. Ouyang, A. Aristov, M. Lelek, X. Hao, and C. Zimmer, "Deep learning massively accelerates super-resolution localization microscopy," *Nat Biotechnol,* vol. 36, no. 5, pp. 460-468, Jun 2018, doi: 10.1038/nbt.4106.

[119] E. Nehme *et al.*, "DeepSTORM3D: dense 3D localization microscopy and PSF design by deep learning," *Nat Methods,* vol. 17, no. 7, pp. 734-740, Jul 2020, doi: 10.1038/s41592-020-0853-5.

[120] Y. Rivenson *et al.*, "Virtual histological staining of unlabelled tissue-autofluorescence images via deep learning," *Nat Biomed Eng,* vol. 3, no. 6, pp. 466-477, Jun 2019, doi: 10.1038/s41551-019-0362-y.

[121] E. M. Christiansen *et al.*, "In Silico Labeling: Predicting Fluorescent Labels in Unlabeled Images," *Cell,* vol. 173, no. 3, pp. 792-803 e19, Apr 19 2018, doi: 10.1016/j.cell.2018.03.040.

[122] C. Ounkomol, S. Seshamani, M. M. Maleckar, F. Collman, and G. R. Johnson, "Label-

free prediction of three-dimensional fluorescence images from transmitted-light microscopy," *Nat Methods,* vol. 15, no. 11, pp. 917-920, Nov 2018, doi: 10.1038/s41592-018-0111-2.

[123] S. M. Guo *et al.*, "Revealing architectural order with quantitative label-free imaging and deep learning," *Elife,* vol. 9, Jul 27 2020, doi: 10.7554/eLife.55502.

[124] S. Cheng *et al.*, "Single-cell cytometry via multiplexed fluorescence prediction by label-free reflectance microscopy," *Sci Adv,* vol. 7, no. 3, Jan 2021, doi: 10.1126/sciadv.abe0431.

[125] J. K. Zhang, Y. He, N. Sobh, and G. Popescu, "Label-free colorectal cancer screening using deep learning and spatial light interference microscopy (SLIM)," *APL Photonics,* vol. 5, no. 4, Apr 2020, doi: 10.1063/5.0004723.

[126] T. Matsumoto *et al.*, "Deep-UV excitation fluorescence microscopy for detection of lymph node metastasis using deep neural network," *Sci Rep,* vol. 9, no. 1, p. 16912, Nov 15 2019, doi: 10.1038/s41598-019-53405-w.

[127] P. Eulenberg *et al.*, "Reconstructing cell cycle and disease progression using deep learning," *Nat Commun,* vol. 8, no. 1, p. 463, Sep 6 2017, doi: 10.1038/s41467-017-00623-3.

[128] O. Schoppe *et al.*, "Deep learning-enabled multi-organ segmentation in whole-body mouse scans," *Nat Commun,* vol. 11, no. 1, p. 5626, Nov 6 2020, doi: 10.1038/s41467-020-19449-7.

[129] Y. Al-Kofahi, A. Zaltsman, R. Graves, W. Marshall, and M. Rusu, "A deep learning-based algorithm for 2-D cell segmentation in microscopy images," *BMC Bioinformatics,* vol. 19, no. 1, p. 365, Oct 3 2018, doi: 10.1186/s12859-018-2375-z.

[130] G. Wu, T. Nowotny, Y. Zhang, H. Q. Yu, and D. D. Li, "Artificial neural network approaches for fluorescence lifetime imaging techniques," *Opt Lett,* vol. 41, no. 11,

pp. 2561-4, Jun 1 2016, doi: 10.1364/OL.41.002561.

[131] L. Heliot and A. Leray, "Simple phasor-based deep neural network for fluorescence lifetime imaging microscopy," *Sci Rep,* vol. 11, no. 1, p. 23858, Dec 13 2021, doi: 10.1038/s41598-021-03060-x.

[132] J. T. Smith *et al.*, "Fast fit-free analysis of fluorescence lifetime imaging via deep learning," *Proc Natl Acad Sci U S A,* vol. 116, no. 48, pp. 24019-24030, Nov 26 2019, doi: 10.1073/pnas.1912707116.

[133] Y. I. Chen *et al.*, "Generative adversarial network enables rapid and robust fluorescence lifetime image analysis in live cells," *Commun Biol,* vol. 5, no. 1, p. 18, Jan 11 2022, doi: 10.1038/s42003-021-02938-w.

[134] C. Szegedy *et al.*, "Going Deeper With Convolutions," presented at the Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015.

[135] X. Z. Kaiming He, Shaoqing Ren, Jian Sun, "Deep Residual Learning for Image Recognition," presented at the Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016.

[136] S. Kiranyaz, O. Avci, O. Abdeljaber, T. Ince, M. Gabbouj, and D. J. Inman, "1D convolutional neural networks and applications: A survey," *Mechanical Systems and Signal Processing,* vol. 151, 2021, doi: 10.1016/j.ymssp.2020.107398.

[137] U. R. Acharya *et al.*, "A deep convolutional neural network model to classify heartbeats," *Comput Biol Med,* vol. 89, pp. 389-396, Oct 1 2017, doi: 10.1016/j.compbiomed.2017.08.022.

[138] O. Avci, O. Abdeljaber, S. Kiranyaz, M. Hussein, and D. J. Inman, "Wireless and real-time structural damage detection: A novel decentralized method for wireless sensor networks," *Journal of Sound and Vibration,* vol. 424, pp. 158-172, 2018, doi: 10.1016/j.jsv.2018.03.008.

[139] M. Bettoni, G. Urgese, Y. Kobayashi, E. Macii, and A. Acquaviva, "A Convolutional Neural Network Fully Implemented on FPGA for Embedded Platforms," presented at the 2017 New Generation of CAS (NGCAS), 2017.

[140] C. Zhang, P. Li, G. Sun, Y. Guan, B. Xiao, and J. Cong, "Optimizing FPGA-based Accelerator Design for Deep Convolutional Neural Networks," in *FPGA '15: Proceedings of the 2015 ACM/SIGDA International Symposium on Field-Programmable Gate Arrays*, 2015, pp. 161-170, doi: https://doi.org/10.1145/2684746.2689060.

[141] A. G. Blaiech, K. Ben Khalifa, C. Valderrama, M. A. C. Fernandes, and M. H. Bedoui, "A Survey and Taxonomy of FPGA-based Deep Learning Accelerators," *Journal of Systems Architecture,* vol. 98, pp. 331-345, 2019, doi: 10.1016/j.sysarc.2019.01.007.

[142] S. C. Smithson, G. Yang, W. J. Gross, and B. H. Meyer, "Neural networks designing neural networks," presented at the Proceedings of the 35th International Conference on Computer-Aided Design, 2016.

[143] H. C. Gerritsen, M. A. Asselbergs, A. V. Agronskaia, and W. G. Van Sark, "Fluorescence lifetime imaging in scanning microscopes: acquisition speed, photon economy and lifetime resolution," *J Microsc,* vol. 206, no. Pt 3, pp. 218-24, Jun 2002, doi: 10.1046/j.1365-2818.2002.01031.x.

[144] Y. Chen, Y. Zhang, D. J. Birch, and A. S. Barnard, "Creation and luminescence of size-selected gold nanorods," *Nanoscale,* vol. 4, no. 16, pp. 5017-22, Aug 21 2012, doi: 10.1039/c2nr30324h.

[145] Y. Zhang, D. J. S. Birch, and Y. Chen, "Two-photon excited surface plasmon enhanced energy transfer between DAPI and gold nanoparticles: Opportunities in intra-cellular imaging and sensing," *Applied Physics Letters,* vol. 99, no. 10, 2011, doi: 10.1063/1.3633066.

[146] G. Wei, J. Yu, J. Wang, P. Gu, D. J. Birch, and Y. Chen, "Hairpin DNA-functionalized gold nanorods for mRNA detection in homogenous solution," *J Biomed Opt,* vol. 21, no. 9, p. 97001, Sep 1 2016, doi: 10.1117/1.JBO.21.9.097001.

[147] Y. Zhang, G. Wei, J. Yu, D. J. Birch, and Y. Chen, "Surface plasmon enhanced energy transfer between gold nanorods and fluorophores: application to endocytosis study and RNA detection," *Faraday Discuss,* vol. 178, pp. 383-94, 2015, doi: 10.1039/c4fd00199k.

[148] G. D. Craig, "Fluorescent Gold Nanorod Probes for the Detection of Cancer mRNA Biomarkers," Doctor of Philosophy, Department of Physics, University of Strathclyde, 2019.

[149] Y. Zhang, J. Yu, D. J. Birch, and Y. Chen, "Gold nanorods for fluorescence lifetime imaging in biology," *J Biomed Opt,* vol. 15, no. 2, p. 020504, Mar-Apr 2010, doi: 10.1117/1.3366646.

[150] M. Castello *et al.*, "A robust and versatile platform for image scanning microscopy enabling super-resolution FLIM," *Nat Methods,* vol. 16, no. 2, pp. 175-178, Feb 2019, doi: 10.1038/s41592-018-0291-9.

[151] S. P. Poland *et al.*, "A high speed multifocal multiphoton fluorescence lifetime imaging microscope for live-cell FRET imaging," *Biomed Opt Express,* vol. 6, no. 2, pp. 277-96, Feb 1 2015, doi: 10.1364/BOE.6.000277.

[152] D. Xiao, Y. Chen, and D. D.-U. Li, "One-Dimensional Deep Learning Architecture for Fast Fluorescence Lifetime Imaging," *IEEE Journal of Selected Topics in Quantum Electronics,* vol. 27, no. 4, pp. 1-10, 2021, doi: 10.1109/jstqe.2021.3049349.

[153] D. Xiao *et al.*, "Dynamic fluorescence lifetime sensing with CMOS single-photon avalanche diode arrays and deep learning processors," *Biomed Opt Express,* vol. 12, no. 6, pp. 3450-3462, Jun 1 2021, doi: 10.1364/BOE.425663.

[154] C. Dong, C. C. Loy, K. He, and X. Tang, "Image Super-Resolution Using Deep Convolutional Networks," *IEEE Trans Pattern Anal Mach Intell,* vol. 38, no. 2, pp. 295-307, Feb 2016, doi: 10.1109/TPAMI.2015.2439281.

[155] W. Yang, X. Zhang, Y. Tian, W. Wang, J.-H. Xue, and Q. Liao, "Deep Learning for Single Image Super-Resolution: A Brief Review," *IEEE Transactions on Multimedia,* vol. 21, no. 12, pp. 3106-3121, 2019, doi: 10.1109/tmm.2019.2919431.

[156] K. Jiang, Z. Wang, P. Yi, and J. Jiang, "A Progressively Enhanced Network for Video Satellite Imagery Superresolution," *IEEE Signal Processing Letters,* vol. 25, no. 11, pp. 1630-1634, 2018, doi: 10.1109/lsp.2018.2870536.

[157] W. Shi *et al.*, "Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 1874-1883.

[158] H. Zhang *et al.*, "High-throughput, high-resolution deep learning microscopy based on registration-free generative adversarial network," *Biomed Opt Express,* vol. 10, no. 3, pp. 1044-1063, Mar 1 2019, doi: 10.1364/BOE.10.001044.

[159] C. Qiao *et al.*, "Evaluation and development of deep neural networks for image super-resolution in optical microscopy," *Nat Methods,* vol. 18, no. 2, pp. 194-202, Feb 2021, doi: 10.1038/s41592-020-01048-5.

[160] L. Fang *et al.*, "Deep learning-based point-scanning super-resolution imaging," *Nat Methods,* vol. 18, no. 4, pp. 406-416, Apr 2021, doi: 10.1038/s41592-021-01080-z.

[161] "Dataset collections in Pytorch, https://pytorch.org/vision/stable/datasets.html# " (accessed 2022).

[162] C. Shorten and T. M. Khoshgoftaar, "A survey on Image Data Augmentation for Deep Learning," *Journal of Big Data,* vol. 6, no. 1, 2019, doi: 10.1186/s40537-019-0197-0.

[163] S. Ross, *Stochastic Processes, 2nd Edition*. Wiley, 1995.

[164] "The Human Protein Atlas, https://www.proteinatlas.org/." (accessed 2022).

[165] "Human Protein Atlas - Single Cell Classification, https://www.kaggle.com/c/hpa-single-cell-image-classification." (accessed 2022).

[166] A. C. Ulku *et al.*, "A 512 × 512 SPAD Image Sensor With Integrated Gating for Widefield FLIM," *IEEE J Sel Top Quantum Electron*, vol. 25, no. 1, pp. 1-12, Jan.-Feb. 2019, Art no. 6801212, doi: 10.1109/JSTQE.2018.2867439.

[167] Y. Zhang, Y. Tian, Y. Kong, B. Zhong, and Y. Fu, "Residual Dense Network for Image Super-Resolution," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 2472-2481.

[168] P. Wang *et al.*, "Understanding Convolution for Semantic Segmentation," arXiv:1702.08502, 2018.

[169] Z. Wang, E. P. Simoncelli, and A. C. Bovik, "Multiscale structural similarity for image quality assessment," in *The Thrity-Seventh Asilomar Conference on Signals, Systems & Computers*, 2003, vol. 2, pp. 1398-1402.

[170] Y. Li *et al.*, "Investigations on Average Fluorescence Lifetimes for Visualizing Multi-Exponential Decays," *Frontiers in Physics,* vol. 8, 2020, doi: 10.3389/fphy.2020.576862.

[171] H. E. Grecco, P. Roda-Navarro, and P. J. Verveer, "Global analysis of time correlated single photon counting FRET-FLIM data," *Opt Express,* vol. 17, no. 8, pp. 6493-508, Apr 13 2009, doi: 10.1364/oe.17.006493.

[172] P. J. Verveer, A. Squire, and P. I. Bastiaens, "Global analysis of fluorescence lifetime imaging microscopy data," *Biophys J,* vol. 78, no. 4, pp. 2127-37, Apr 2000, doi: 10.1016/S0006-3495(00)76759-2.

[173] S. Z. Li, *Markov Random Field Modeling in Image Analysis*. Springer London, 2009.

[174] C. M. Bishop and N. M. Nasrabadi, *Pattern recognition and machine learning*. New York: Springer, 2006.

[175] A. Trockman and J. Z. Kolter, "Patches Are All You Need?," *arXiv:2201.09792,* 2022.

[176] Y. Chen, S. Liu, and X. Wang, "Learning continuous image representation with local implicit image function.," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 8628–8638.

[177] K. Genova, F. Cole, A. Sud, A. Sarna, and T. Funkhouser, " Local deep implicit functions for 3d shape," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 4857–4866.

[178] C. Jiang, A. Sud, A. Makadia, J. Huang, M. Nießner, and T. Funkhouser, " Local implicit grid representations for 3d scenes," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 6001–6010.

[179] V. Sitzmann, J. N. P. Martel, A. W. Bergman, D. B. Lindell, and G. Wetzstein, "Implicit Neural Representations with Periodic Activation Functions," *arXiv:2006.09661,* 2020.

[180] V. Sitzmann, M. Zollhöfer, and G. Wetzstein, "Scene Representation Networks: Continuous 3D-Structure-Aware Neural Scene Representations," *arXiv:1906.01618,* 2019.

[181] J. Tang, X. Chen, and G. Zeng, "Joint Implicit Image Function for Guided Depth Super-Resolution," presented at the Proceedings of the 29th ACM International Conference on Multimedia, 2021.

[182] A. Adan, G. Alizada, Y. Kiraz, Y. Baran, and A. Nalbant, "Flow cytometry: basic principles and applications," *Crit Rev Biotechnol,* vol. 37, no. 2, pp. 163-176, Mar 2017, doi: 10.3109/07388551.2015.1128876.

[183] M. G. Macey, *Flow Cytometry*. 2007.

[184] K. M. McKinnon, "Flow Cytometry: An Overview," *Curr Protoc Immunol,* vol. 120, pp. 5 1 1-5 1 11, Feb 21 2018, doi: 10.1002/cpim.40.

[185] S. C. De Rosa, J. M. Brenchley, and M. Roederer, "Beyond six colors: a new era in flow cytometry," *Nat Med,* vol. 9, no. 1, pp. 112-7, Jan 2003, doi: 10.1038/nm0103-112.

[186] A. Cossarizza *et al.*, "Guidelines for the use of flow cytometry and cell sorting in immunological studies (second edition)," *Eur J Immunol,* vol. 49, no. 10, pp. 1457-1973, Oct 2019, doi: 10.1002/eji.201970107.

[187] J. P. Houston, Z. Yang, J. Sambrano, W. Li, K. Nichani, and G. Vacca, "Overview of fluorescence lifetime measurements in flow cytometry," in *Flow cytometry protocols*. New York, NY: Humana Press, 2018, pp. 421-446.

[188] B. G. Pinsky, J. J. Ladasky, J. R. Lakowicz, K. Berndt, and R. A. Hoffman, "Phase-resolved fluorescence lifetime measurements for flow cytometry," *Cytometry,* vol. 14, no. 2, pp. 123-35, 1993, doi: 10.1002/cyto.990140204.

[189] J. A. Steinkamp and J. F. Keij, "Fluorescence intensity and lifetime measurement of free and particle-bound fluorophore in a sample stream by phase-sensitive flow cytometry," *Review of Scientific Instruments,* vol. 70, no. 12, pp. 4682-4688, 1999, doi: 10.1063/1.1150143.

[190] C. Deka, B. E. Lehnert, N. M. Lehnert, G. M. Jones, L. A. Sklar, and J. A. Steinkamp, "Analysis of fluorescence lifetime and quenching of FITC-conjugated antibodies on cells by phase-sensitive flow cytometry," *Cytometry,* vol. 25, no. 3, pp. 271-279, 1996, doi: 10.1002/(sici)1097-0320(19961101)25:3<271::Aid-cyto8>3.0.Co;2-i.

[191] M. Suzuki *et al.*, "A high-throughput direct fluorescence resonance energy transfer-based assay for analyzing apoptotic proteases using flow cytometry and fluorescence

lifetime measurements," *Anal Biochem,* vol. 491, pp. 10-7, Dec 15 2015, doi: 10.1016/j.ab.2015.08.022.

[192] A. V. Gohar, R. Cao, P. Jenkins, W. Li, J. P. Houston, and K. D. Houston, "Subcellular localization-dependent changes in EGFP fluorescence lifetime measured by time-resolved flow cytometry," *Biomed Opt Express,* vol. 4, no. 8, pp. 1390-400, 2013, doi: 10.1364/BOE.4.001390.

[193] W. Li, G. Vacca, M. Castillo, K. D. Houston, and J. P. Houston, "Fluorescence lifetime excitation cytometry by kinetic dithering," *Electrophoresis,* vol. 35, no. 12-13, pp. 1846-54, Jul 2014, doi: 10.1002/elps.201300618.

[194] J. Nedbal *et al.*, "Time-domain microfluidic fluorescence lifetime flow cytometry for high-throughput Forster resonance energy transfer screening," *Cytometry A,* vol. 87, no. 2, pp. 104-18, Feb 2015, doi: 10.1002/cyto.a.22616.

[195] J. Leonard *et al.*, "High-throughput time-correlated single photon counting," *Lab Chip,* vol. 14, no. 22, pp. 4338-43, Nov 21 2014, doi: 10.1039/c4lc00780h.

[196] D. Kage, K. Hoffmann, M. Wittkamp, J. Ameskamp, W. Gohde, and U. Resch-Genger, "Luminescence lifetime encoding in time-domain flow cytometry," *Sci Rep,* vol. 8, no. 1, p. 16715, Nov 13 2018, doi: 10.1038/s41598-018-35137-5.

[197] D. Tyndall *et al.*, "A high-throughput time-resolved mini-silicon photomultiplier with embedded fluorescence lifetime estimation in 0.13 mum CMOS," *IEEE Trans Biomed Circuits Syst,* vol. 6, no. 6, pp. 562-70, Dec 2012, doi: 10.1109/TBCAS.2012.2222639.

[198] S. Cova, M. Ghioni, A. Lacaita, C. Samori, and F. Zappa, "Avalanche photodiodes and quenching circuits for single-photon detection," *Appl Opt,* vol. 35, no. 12, pp. 1956-76, Apr 20 1996, doi: 10.1364/AO.35.001956.

[199] B. E. A. Saleh and M. C. Teich, *Fundamentals of Photonics* (Wiley Series in Pure and Applied Optics). 1991.

[200] R. Widenhorn, A. Dupret, I. Vornicu, R. Carmona-Galán, and Á. Rodríguez-Vázquez, "A SPAD-based 3D imager with in-pixel TDC for 145ps-accuracy ToF measurement," presented at the Image Sensors and Imaging Systems 2015, 2015.

[201] C. Bruschini, H. Homulle, I. M. Antolovic, S. Burri, and E. Charbon, "Single-photon avalanche diode imagers in biophotonics: review and outlook," *Light Sci Appl,* vol. 8, p. 87, 2019, doi: 10.1038/s41377-019-0191-5.

[202] "https://www.picoquant.com/rapidFLIM." (accessed 2022).

[203] J. Martini, V. Andresen, and D. Anselmetti, "Scattering suppression and confocal detection in multifocal multiphoton microscopy," *J Biomed Opt,* vol. 12, no. 3, p. 034010, May-Jun 2007, doi: 10.1117/1.2736425.

[204] S. Kumar *et al.*, "Multifocal multiphoton excitation and time correlated single photon counting detection for 3-D fluorescence lifetime imaging," *Opt Express,* vol. 15, no. 20, pp. 12548-61, Oct 1 2007, doi: 10.1364/oe.15.012548.

[205] S. Zhou, Y. Wu, Z. Ni, X. Zhou, H. Wen, and Y. Zou, "DoReFa-Net: Training Low Bitwidth Convolutional Neural Networks with Low Bitwidth Gradients," *arXiv:1606.06160* 2016.

[206] P. Yin, J. Lyu, S. Zhang, S. Osher, Y. Qi, and J. Xin, "Understanding Straight-Through Estimator in Training Activation Quantized Neural Nets," *arXiv:1903.05662,* 2019.

[207] "https://www.xilinx.com/products/silicon-devices/fpga/what-is-an-fpga.html." (accessed 2022).

[208] Xilinx. "Vivado Design Suite Tutorial High-Level Synthesis, https://docs.xilinx.com/v/u/en-US/ug871-vivado-high-level-synthesis-tutorial." (accessed 2022).

[209] D. Xiao, Z. Zang, Q. Wang, Z. Jiao, F. M. Della Rocca, Y. Chen, and D. Li, "Smart Wide-Field Fluorescence Lifetime Imaging System with CMOS Single-Photon

Avalanche Diode Arrays", 44th Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC 2022), Glasgow, UK