HAMAMATSU PRESENTS

ANALYTICAL TALKS

WATCH NOW

Received: 15 November 2019 | Revised: 22 February 2020 | Accepted: 10 March 2020

DOI: 10.1002/jbio.201960186

REVIEW ARTICLE

JOURNAL OF BIOPHOTONICS

Deep learning a boon for biophotonics?

Pranita Pradhan[1,2] | **Shuxia Guo**[1,2] | **Oleg Ryabchykov**[1,2] | **Juergen Popp**[1,2] | **Thomas W. Bocklitz**[1,2]

[1] Institute of Physical Chemistry and Abbe Center of Photonics, Friedrich-Schiller-University, Jena, Germany

[2] Leibniz Institute of Photonic Technology (Leibniz-IPHT), Member of Leibniz Research Alliance 'Health Technologies', Jena, Germany

Correspondence

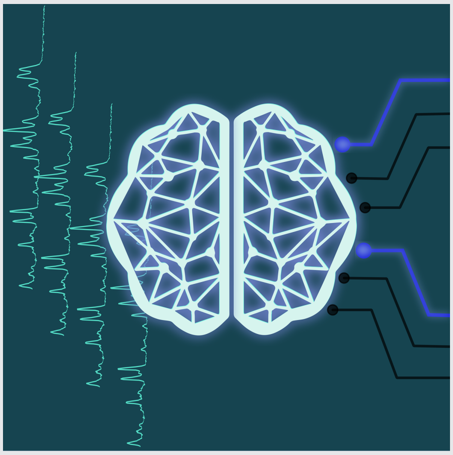
Thomas W. Bocklitz, Institute of Physical Chemistry and Abbe Center of Photonics (IPC), Friedrich-Schiller-University, Helmholtzweg 4, D-07743 Jena, Germany.
Email: thomas.bocklitz@uni-jena.de

Funding information

Bundesministerium für Bildung und Forschung, Grant/Award Number: Uro-MDD (FKZ 03ZZ0444J); Deutsche Forschungsgemeinschaft, Grant/Award Numbers: BO 4700/1-1, BO 4700/4-1, PO 563/30-1, STA 295/11-1; Thüringer Ministerium für Wirtschaft, Wissenschaft und Digitale Gesellschaft, Grant/Award Number: DigLeben (5575/10-9)

Abstract

This review covers original articles using deep learning in the biophotonic field published in the last years. In these years deep learning, which is a subset of machine learning mostly based on artificial neural network geometries, was applied to a number of biophotonic tasks and has achieved state-of-the-art performances. Therefore, deep learning in the biophotonic field is rapidly growing and it will be utilized in the next years to obtain real-time biophotonic decision-making systems and to analyze biophotonic data in general. In this contribution, we discuss the possibilities of deep learning in the biophotonic field including image classification, segmentation, registration, pseudostaining and resolution enhancement. Additionally, we discuss the potential use of deep learning for spectroscopic data including spectral data preprocessing and spectral classification. We conclude this review by addressing the potential applications and challenges of using deep learning for biophotonic data.



KEYWORDS

artificial neural networks, biophotonics, deep learning, spectroscopy

1 | INTRODUCTION

Biophotonics is a rapidly growing multidisciplinary field that utilizes the interaction of light with biological systems and investigates these biological systems at the cellular, molecular and tissue level. Since the past decade, these biophotonic technologies are globally established in biotechnology companies, healthcare organizations, medical instrument suppliers and pharmaceutical manufacturers. For instance, laser-based therapy is an important part of medical sciences today, and is used for light-guided therapies in various organs. Other light-based technologies like multiphoton microscopy (MPM), optical coherence tomography (OCT), Raman spectroscopy, infrared spectroscopy (IR), photoacoustic imaging (PAI) and fluorescence life-time imaging microscopy (FLIM) are further useful tools in biomedical and biophotonic research [1, 2]. For example, nonlinear multimodal imaging which includes two-photon excited fluorescence microscopy (TPEF), second-harmonic generation (SHG) and coherent anti-stokes Raman scattering (CARS), is widely used in dermatology, physiology, neurobiology and embryology. Similarly, technologies like OCT are mainly used in ophthalmology and cardiology, while spectroscopic techniques have various clinical and pharmaceutical applications.

Nowadays, biophotonic technologies are witnessing a rapid development in the instrumentation of the optical devices which is fastening the imaging speed, increasing the penetration depth and enhancing the resolution of the optical images. These developments make it possible to measure label-free molecular information of samples like cells or tissue. As all of the biophotonic technologies are label-free, the spectral and image data is untargeted. That means it is difficult to interpret a specific contrast associated to a chemical structure or a biomolecule in biophotonic data. Therefore, the interpretation of biophotonic data has to be generated using appropriate analysis techniques like statistics, chemometrics or machine learning. Additionally, the technical improvement of these biophotonic technologies has given rise to large datasets, which require big data analysis methods to be applied to biophotonic data [3]. Overall, interpreting and handling biophotonic data are two obvious challenges for the biophotonic community (Table 1).

TABLE 1 List of mathematical symbols

Symbol	Explanation
x	An input as a scalar (integer or real)
$\mathbf{x}$	An input vector or 1D data
$\mathbf{X}$	An input matrix or 2D data
Θ	A set of trainable parameters
$\mathcal{W}$	A set of weights
$\mathcal{B}$	A set of biases
L	Number of layers in a model
l	Index of L layers
η	Learning rate
IR^D	A D dimensional set of real numbers
IR^N	A N dimensional set of real numbers
τ	An index for iteration
P	A probability distribution over discrete variable
χ	An input space representation
$\mathcal{F}$	A latent space representation
E	Loss function
$\mathbb{E}_{x,y}[f(x)]$	Expectation of $f(x)$ with respect to $P(y\|x)$

In this context, the well-established statistical pattern-recognition methods are employed which extract "features" or "patterns" from the biophotonic data. These techniques are called "feature extraction" methods. Feature extraction is a process of dimension reduction used to transform high dimensional data to low dimensional data. Subsequently, the low dimensional data commonly called "features" can be used to construct learning algorithms. This procedure is shared by most of the machine learning algorithms where feature extraction is followed by prediction of the outcome or probabilities [4]. Classification or regression models are common examples of machine learning algorithms where features from images (like shape, texture, color features) or features of spectra (like intensity values at specific wavenumbers in Raman spectroscopy) are extracted to construct a predictive model. These machine-learning algorithms in combination with a high computational power can be utilized to interpret the biophotonic data. A subset of machine learning algorithms is called "deep learning," which requires least manual intervention for feature extraction and can be employed as a decision-making algorithm with high accuracy. Since a decade, deep learning algorithms have achieved promising results in clinical radiology covering a wide range of applications from cancer diagnosis to personalized therapies [5]. Similar to clinical radiology, introduction of deep learning algorithms in biophotonics has also revolutionized the data analysis in this field. The respective research will be further discussed in this article.

This review article aims to give an overview of deep learning techniques for spectroscopic data intended for the multidisciplinary readership of *J. Biophotonics*. We aim to stimulate the interest of researchers and data scientists to foster applications of deep learning in biophotonics by discussing the ongoing evolution in the field of biophotonics and deep learning. Additionally, we emphasize potential applications and challenges encountered while applying deep learning for biophotonic data. We structure our article in the following manner: section 2 discusses the commonly used deep learning architectures to analyze biophotonic data. Section 3 presents the applications of deep learning for preprocessing, classifying and segmenting microscopic imaging data. Section 4 presents the preprocessing and analysis for spectroscopic data using deep learning. Further, section 5 addresses the challenges faced by a researcher while analyzing biophotonic data using deep learning and we introduce the approaches for overcoming these challenges. Lastly, we conclude our review in section 6 by answering the question "Is deep learning a boon for biophotonics?"

2 | DEEP LEARNING—AN OVERVIEW

With rising complexity of spectroscopic datasets and the need to achieve good decision-making systems, more advanced machine learning algorithms are required. Briefly, a machine-learning algorithm is an algorithm that is able to learn from data. A special kind of machine learning algorithms is deep learning algorithm. A deep learning algorithm is based on four major components which are an optimization algorithm, a cost function, a dataset and a deep learning model. Shortly, an optimization algorithm is an iterative method to compare various solutions for a problem until an optimal solution is obtained. A cost function is a mathematical formula used to evaluate the performance of a deep learning model. A dataset is one of the major components for training the deep learning models and can be split into three parts: training, validation and testing dataset. The training dataset is used for training the deep learning model, the validation dataset is used to tune the hyperparameters of the deep learning model and the independent test dataset or holdout set is used to evaluate the performance of the model in an unbiased manner [4, 6–8]. The last necessary component is a deep learning model which is made of a series of layers and hyperparameters depending on various architectures, which are discussed in the further course of the section.

Deep learning algorithms have widespread applications in speech recognition, natural language processing, healthcare and so on. Particularly in healthcare, deep learning is often applied to radiology data. Similar to clinical radiology, traditional artificial neural networks [9] were applied since the 1990s to biophotonic data [10, 11] the recently developed deep learning models especially convolutional neural networks have achieved state-of-the art performance in the biophotonic field. This section summarizes a few deep learning models that are commonly used to analyze spectroscopic data. Each subsection gives a brief overview of a specific deep learning architecture combined with an illustration of how to apply these deep learning architectures for image and spectral data.

2.1 | Feed-forward neural network

Feed-forward neural network commonly called artificial neural network (ANN) or multilayer perceptron (MLP) [9, 12, 13] are the basis of most of the deep learning models utilized today. MLPs are loosely inspired by the human neural system. These models are called feed-forward neural network as the input flows only in the forward direction without a feedback from the output into the model. Specifically, a feed-forward neural network passes the input $\mathbf{x} = \{x_i\} \in \mathrm{IR}^D$, through a series of neurons with an activation a and a set of trainable parameters $\Theta = \{\mathcal{W}, \mathcal{B}\}$ to obtain an output y. An activation function $a = \sigma(\mathbf{w}^T\mathbf{x} + b)$ introduces an elementwise nonlinearity $\sigma(.)$ to the output of a neuron, which is a linear combination of the neuron's input and the parameters Θ (see Figure 1). A composition of many such transformations forms the basis of a feed-forward neural network where the input is passed through a series of "hidden layers" to obtain the output. A neuron output y_k of an MLP with M and D neurons in two hidden layers l and $l-1$ respectively, can be represented as

$$y_k(\mathbf{x};\Theta) = \sigma^l\left(\Sigma_{j=1}^{M} W_{kj}^{l} \sigma^{l-1}\left(\Sigma_{i=1}^{D} W_{ji}^{l-1} x_i + b_j^{l-1}\right) + b_k^{l}\right), \tag{1}$$

where Θ is the set of trainable parameters, W_{ji} is a weight matrix of size $j \times i$, with i inputs and j activations of $(l-1)$th layer. During the training of a feed-forward neural network, the model parameters Θ are iteratively updated using an optimizer until convergence is achieved. A stochastic gradient descent (SGD) optimizer is commonly used in the literature [15, 16], which performs typically the minimization of a loss or a cost function E by the

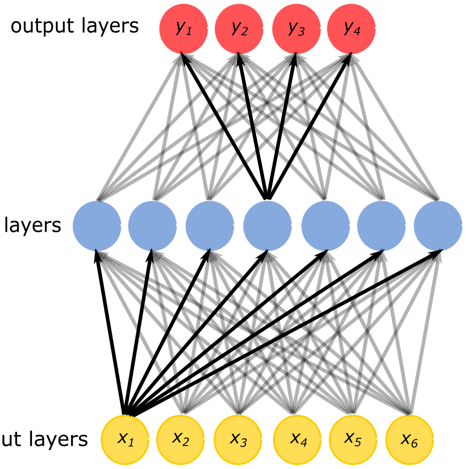


FIGURE 1 A feed-forward neural network or a multilayer perceptron with an input $\mathbf{x} \in \mathrm{IR}^D$, $D = 6$ and output $y \in \mathrm{IR}^N$, $N = 4$ is shown. The input to the network (depicted in yellow) can be features (like histogram features, local binary patterns [14]) obtained from an image or features (like intensity values of different wavenumbers) obtained from a spectrum which passes through the neurons of the hidden layer depicted in blue. The connections between the neurons are weighted by $\mathcal{W}$ and the data is further passed through the layers with activation function a to obtain an output shown in red. The weights are updated using back-propagation as explained in Section 2.1

back-propagation method [17, 18]. Back-propagation minimizes the loss function in the parameter space Θ by computing a gradient of the loss function $E(\Theta)$ [17]. Based on the gradient of the loss function $\nabla E(\Theta)$ computed for all the layers, the model parameters $\Theta = \{\mathcal{W}, \mathcal{B}\}$ can be updated in each iteration τ using the formula given below:

$$\Theta^{(\tau+1)} = \Theta^{(\tau)} - \eta \nabla E(\Theta)^{(\tau)} \; . \tag{2}$$

Here, τ represents an iteration index and η is the learning rate. In addition to the SGD optimizer, other optimizers like Adam [19], Adadelta [20] and Adagrad [21] have also been reported in the literature.

MLPs have widespread applications in image and spectral classification as illustrated in Figure 2. The figure shows an MLP that utilizes image features or spectral features as the input. These features are further propagated through the network to emerge at the output neuron as class outputs (see Figure 2). The class outputs can be tumor/normal for a diagnostic task, disease stages for a disease assessment task or the type of pollen grains for a classification task of pollen grains. Mostly MLPs require the extraction of features from image or spectral data, which is one of the limitations of these basic neural networks. Therefore, more advanced deep learning architectures like convolutional neural networks are required.

2.2 | Convolutional neural network

A convolutional neural network (CNN) [22] is a variant of a MLP, which can work on grid data, for instance spectra or images. Unlike MLPs, CNNs consider the spatial information of an image or temporal/spectral information of a signal directly. This is achieved by convolving the input, like an image $\mathbf{X}$, with trainable kernels or weights $\mathbf{W}_k$ to generate a feature map $\mathbf{X}_k$. Mathematically, a feature map $\mathbf{X}_k$ for the lth layer of a CNN is given by

$$\mathbf{X}_k^l = \sigma\left(\mathbf{W}_k^{l-1} * \mathbf{X}_k^{l-1} + b_k^{l-1}\right) \; , \tag{3}$$

where $\mathcal{W} = \{\mathbf{W}_1, \mathbf{W}_2, ..., \mathbf{W}_K\}$ are K trainable kernels and $\mathcal{B} = \{b_1, b_2, ..., b_K\}$ are the biases. The illustration of a CNN architecture in Figure 3 shows a kernel $\mathbf{W}_1$ of size 3×3, which is convolved with an image $\mathbf{X}$ in a raster pattern with a stride value of 1 pixel (first layer). This forms a feature map or a linearly convolved image $\mathbf{X}_1$. The linearly convolved image is further subjected to an elementwise nonlinear transformation σ which is typically a rectified linear unit (ReLU) [23], tanh [24] and sigmoid [25] function. The activation function σ is important in CNNs to introduce a nonlinearity to the model. Generally, a softmax activation function [6] in the last layer of a model utilized for classification tasks is used. The softmax activation layer maps the activations of the final layer to a probability distribution of classes $P(y|\mathbf{X};\Theta)$ given as

$$P(y|\mathbf{X};\Theta) = softmax(\mathbf{X};\Theta) = \frac{e^{\left(\mathbf{W}_i^l\right)^T \mathbf{X} + b_i^l}}{\sum_{c=1}^{C} e^{\left(\mathbf{W}_c^l\right)^T \mathbf{X} + b_c^l}} \; , \tag{4}$$

where $\mathbf{W}_i^l$ and $\mathbf{b}_i^l$ are the kernel and bias of the lth layer leading to a normalized probability distribution of class i. In contrast to other traditional activation functions, the output of a softmax activation function is normalized

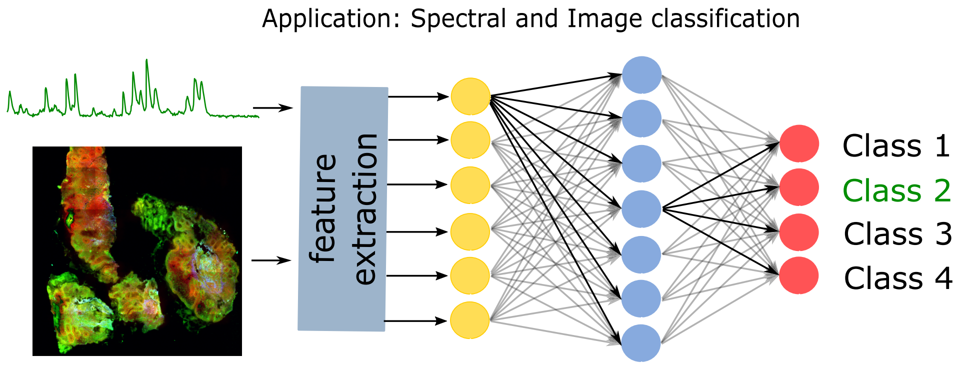
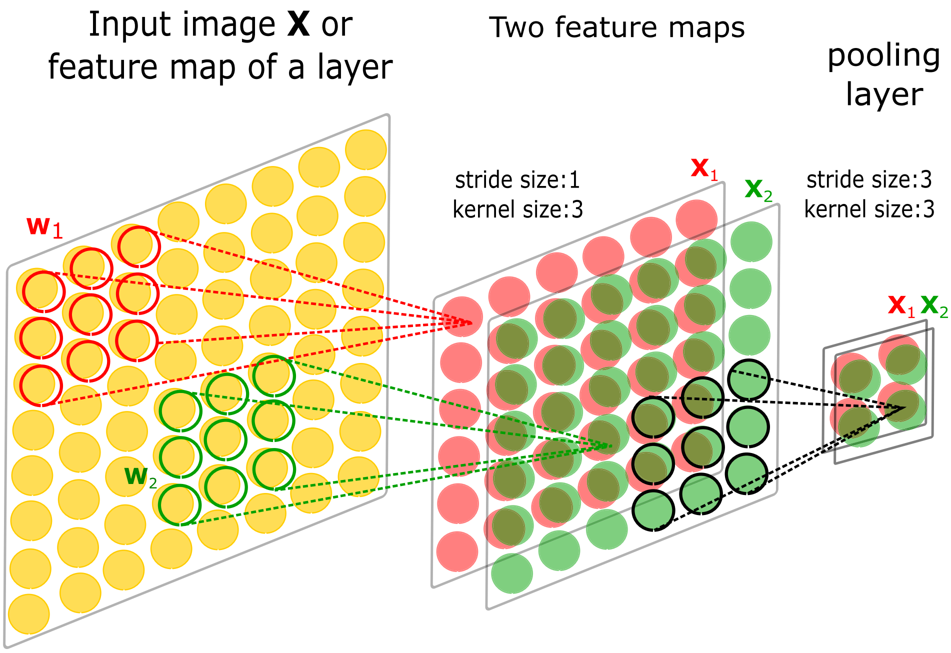


FIGURE 2 Applications of MLPs are shown, where each input neuron utilizes the features obtained from a Raman spectrum (top) and nonlinear multimodal image (bottom). The nonlinear multimodal image is composed of CARS signal as red channel, TPEF signal as green channel and SHG signal as blue channel. The input vector at the first layer is a vector of image features or spectral features. The output neuron of the MLP is a label or a class probability of the input spectrum or of the input image. CARS, coherent anti-stokes Raman scattering; MLP, multilayer perceptron; SHG, second-harmonic generation; TPEF, two-photon excited fluorescence microscopy

FIGURE 3 A general structure of a convolution neural network (CNN) is shown. The input image **X** or a feature map of a layer is convolved by two kernels $\mathbf{W}_1$ and $\mathbf{W}_2$. Each kernel of size 3 × 3 is convolved with a small section of the input image and is shifted with a stride of 1 pixel (first layer) in a raster pattern to obtain a whole feature map $\mathbf{X}_1$ and $\mathbf{X}_2$. The figure also shows a pooling layer of a CNN, which condenses the spatial information of the feature maps making CNNs computationally efficient



between 0 and 1, and the sum of all outputs is equal to 1. A softmax activation function can be used as a last layer for both CNN and MLP in classification tasks. Similar to MLPs, back-propagation in CNNs is performed to update the weights in each kernel, which are computed using the gradients of the loss function determined in forward pass.

Unlike MLPs, CNNs utilize three other important concepts including weight sharing, pooling layers and receptive field (see Figure 3). A weight sharing reduces the number of parameters by sharing weights for all neurons in a feature map. Pooling layers aggregate the neighboring pixel values to reduce the spatial dimension of the input images or the feature maps. A receptive field is a region in the input space that is affected by a kernel. The pixels of an image closer to the center of the receptive field contribute more to the output feature [6].

CNNs are immensely used in biophotonics for image and spectrum classification (see Figure 4), disease characterization and microorganism identification. These applications are further explained in section 3 and section 4. CNNs are also used in other deep learning architectures like auto-encoders and generative adversarial networks explained in section 2.4 and section 2.5, respectively.

2.3 | Recurrent neural network

Standard neural networks like MLPs have certain limitations while working with sequence data like spectroscopic data or time series. One of the limitations is that MLPs fail to consider the entire history of a sequenced input vector for obtaining an output [28] whereas, recurrent neural networks (RNNs) [17] incorporate neurons that span the input over time. Moreover, RNNs have hidden layers that add memory to the network over time.

RNNs can have three types of architectures to solve the sequence data problem: (a) the one-to-many RNN architecture has one input neuron and a sequenced or many output neurons, which are used for image captioning [29], (b) the many-to-one RNN architecture comprises a sequenced or many input neurons and one output neuron, which is used for text classification [30] and lastly (c) the many-to-many RNN architecture has a sequenced or many input neurons and a sequenced or many output neurons, which is mostly used for machine translation [31]. In addition to the earlier mentioned applications, RNNs have obtained promising results in natural language processing, speech recognition and machine translation tasks [32]. Moreover, a recent study reported the use of RNNs for the analysis of genetic data [33]. Despite the enormous development of RNNs, they are underexplored in the field of biophotonics as compared to MLPs and CNNs. Nevertheless, RNNs can build intelligent systems and its use in spectrum preprocessing, wavenumber calibration or intensity calibration, spectrum classification, decoding biomolecular markers from bio-spectroscopic data, learning spatial-spectral-temporal features for spectral data and phase retrieval of nonlinear optical spectroscopic data can be investigated in the future.

A typical many-to-many RNN structure is shown in Figure 5. The figure shows three unit types, an input vector, a hidden state vector and an output vector. For

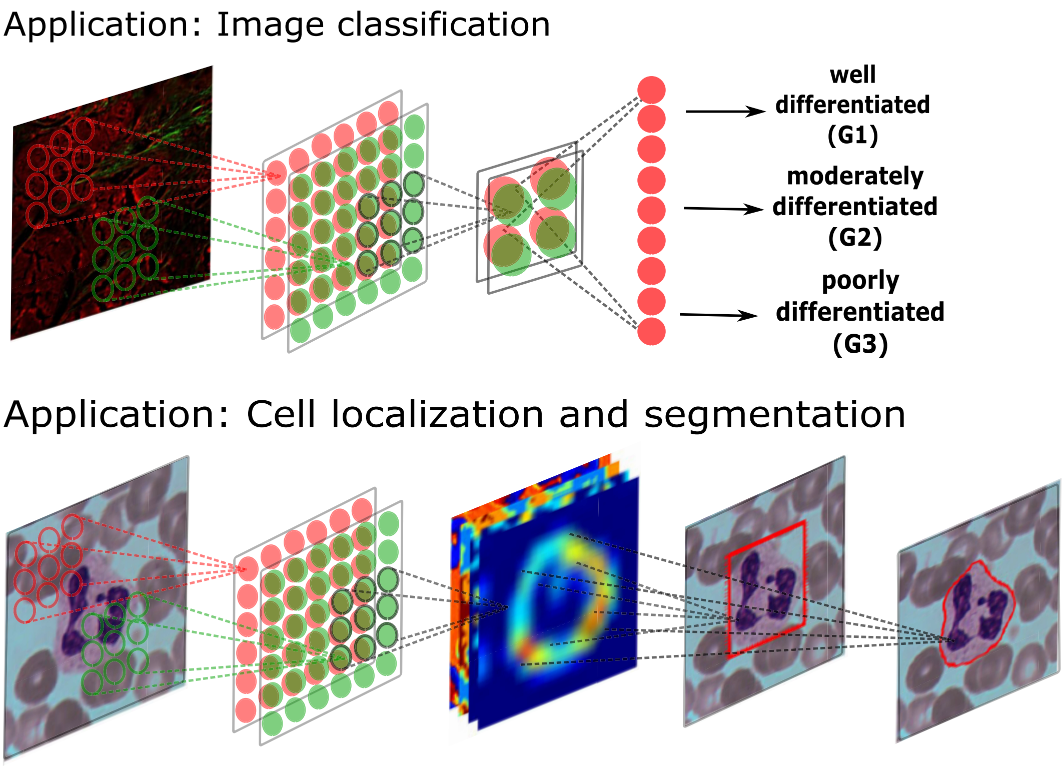

Application: Image classification
well differentiated (G1)
moderately differentiated (G2)
poorly differentiated (G3)
Application: Cell localization and segmentation

FIGURE 4 This figure shows application of CNNs like image classification (upper panel), localization and segmentation (lower panel). For image classification, a multiphoton image is used for classifying three grades of hepatocellular carcinoma (upper panel). For cell localization task, a leukocyte mask was generated using a CNN to localize and segment leukocytes in blood smear images (lower panel). These images are reproduced and modified from references [26, 27]. CNN, convolution neural network

sequenced input data ($\mathbf{x}_1$, $\mathbf{x}_2$, $\mathbf{x}_3$,..., $\mathbf{x}_T$), an RNN can have many outputs ($\mathbf{y}_1$, $\mathbf{y}_2$, $\mathbf{y}_3$,..., $\mathbf{y}_{T+N}$) or the same number of outputs like the input data ($\mathbf{y}_1$, $\mathbf{y}_2$, $\mathbf{y}_3$,..., $\mathbf{y}_T$) or just one output unit y. The intermediate layer represents the hidden state of the RNN. The hidden state $\mathbf{h}_t$ is the memory of the network and is calculated using the hidden state of the previous step $\mathbf{h}_{(t-1)}$ and the input vector at the current step $\mathbf{x}_t$. The hidden state at the first time step is initialized with zeros

$$\mathbf{h}_t = 0 \text{ for } t = 0. \tag{5}$$

The hidden state for the intermediate time steps is calculated by

$$\mathbf{h}_t = \sigma(\mathbf{W}\mathbf{x}_t + \mathbf{U}\mathbf{h}_{t-1} + b) \text{ for } t \neq 0. \tag{6}$$

Here, $\mathbf{U}$ is the weight vector of the hidden layer, $\mathbf{W}$ is the weight vector of the input layer and $\mathbf{V}$ is the weight vector of the output layer shared over time (see Figure 5). Applications of many-to-many RNNs for spectra preprocessing, where the input vector for the many-to-many RNN is a raw spectrum and the output vector is a preprocessed spectrum, still requires investigation. However, many-to-one and many-to-many RNNs can also be used for classification purposes. In such cases, a softmax activation layer is added to the output sequence of the RNN model in order to achieve posterior probabilities for the classes.

Nevertheless, standard RNNs report some shortcomings. Firstly, RNNs require higher computational power and larger training data than usual CNNs. A standard RNN calculates an output at each time step utilizing just the past and the present element of the input vector. For spectroscopic data, the past, present and future states (or wavenumbers) of the spectra influence the output at a particular time step, and the application of bidirectional RNNs can be investigated. A bidirectional RNN utilizes hidden states from opposite directions to update the output sequence at a particular time step. Another shortcoming of RNNs is the problem of vanishing gradients, which occurs due to the deep structure of RNNs. To circumvent this problem, other variations of RNN including long short-term memory (LSTM) and gated recurrent unit (GRU) networks are used and have achieved better performances [34]. A comprehensive discussion of the variations of RNNs is out of scope of this review.

2.4 | Auto-encoder

Auto-encoders (AE) [35, 36] are ANNs consisting of two parts: an encoder and a decoder. The encoder transforms a D dimensional input $\mathbf{x} \in \mathrm{IR}^D = \chi$ to a N dimensional hidden states $\mathbf{h} \in \mathrm{IR}^N = \mathcal{F}$, $N < D$, where χ is the input space and $\mathcal{F}$ is the latent space representation. The latent space $\mathcal{F}$ is represented by the bottleneck of the model (see Figure 6). The bottleneck layer compresses the input space representation χ to capture the most salient features of the input data. The representation of the hidden states $\mathbf{h}$ in the bottleneck layer can be written as

$$\mathbf{h} = \sigma(\mathbf{W}\mathbf{x} + \mathbf{b}). \quad (7)$$

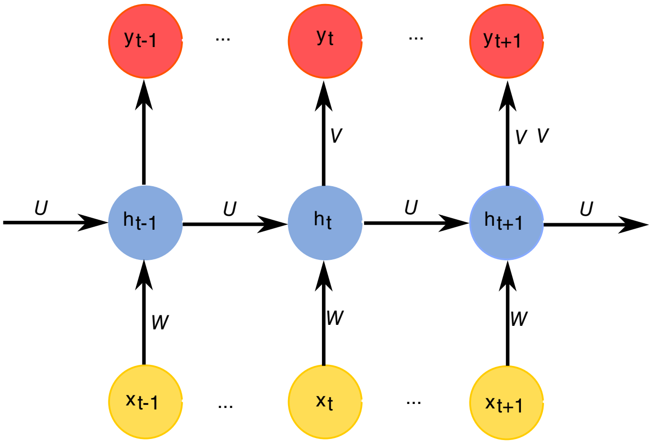


FIGURE 5 A structure of recurrent neural network is shown. A set of sequenced data x with T time steps is given as an input (yellow) to reconstruct a sequenced output vector y (red) with equal number of time steps. The hidden states (blue) store the features or act as memory unit of the RNN network. The weight matrices W,U, V are updated during the training of RNNs

The dimension of the bottleneck layer is smaller as compared to the dimensions of the input layer to avoid the encoder from learning an identity function.

A decoder transforms the bottleneck features of the hidden states $\mathbf{h}$ back to a reconstructed input $\mathbf{x}'$ of the same dimension as $\mathbf{x}$. The reconstructed input $\mathbf{x}'$ can be given as

$$\mathbf{x}' = \sigma(\mathbf{W}'\mathbf{h} + \mathbf{b}'). \quad (8)$$

Here, $\mathbf{W}'$ and $\mathbf{b}'$ are the weight matrix and bias of the decoder respectively. The training of an auto-encoder is performed through back-propagation of reconstruction error calculated between the original and the reconstructed input.

Traditionally, auto-encoders were used for dimensionality reduction [6]. Simple auto-encoders find its application for denoising, image deblurring and semantic segmentation (see Figure 7), which will be discussed in section 3 [39]. Additionally, variations of auto-encoders like stacking auto-encoder, sparse auto-encoder, denoising auto-encoder, convolutional auto-encoder, variational auto-encoder and contractive auto-encoder are used to prevent the learning of an identity function by the encoder, as stated earlier [40]. Moreover, auto-encoders can be a part of adversarial networks discussed in section 2.5.

2.5 | Generative adversarial network

A generative adversarial network (GAN) [41] is a special type of ANN that consists of two networks: a generator and a discriminator, which are trained simultaneously. The input to the generator is either a random noise vector $\mathbf{z}$ or a real data, like an image $\mathbf{X}$, sampled from a prior distribution p_{data}. The generator is

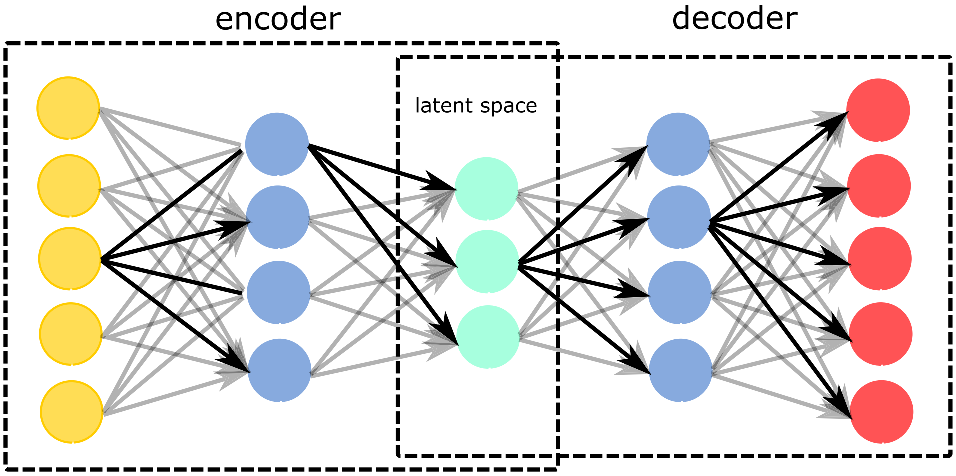


FIGURE 6 An auto-encoder (AE) structure with two parts, an encoder and a decoder, is shown. An encoder transforms the input information (shown in yellow) into a latent space representation (shown in cyan) $\mathcal{F}$, which is transferred by the decoder to reconstruct an output (shown in red) in the same space representation as the input. Both the parts can be constructed using a CNN or an MLP. CNN, convolution neural network; MLP, multilayer perceptron

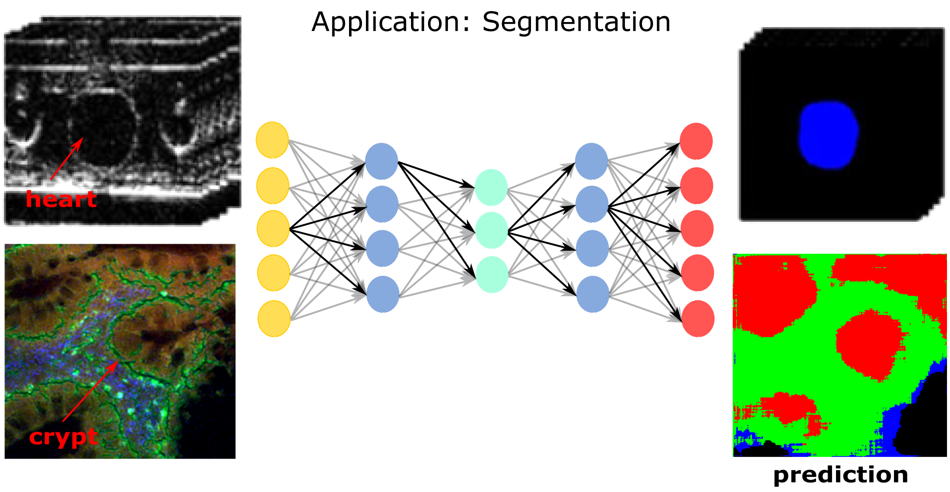


FIGURE 7 Applications of auto-encoders are shown in this figure. The upper panel shows a segmentation of a Drosophila heart, where optical coherence tomography images are used as an input to the AE model and the output is a segmented drosophila heart. Similarly, the lower panel shows a segmented crypt in a nonlinear multimodal image. The nonlinear multimodal image is used as an input to the AE model to obtain a false-color image with four distinct regions (crypt region in red) at the output. Similar to Figure 2, the nonlinear multimodal image is composed of CARS, TPEF and SHG signal. The images are reprinted from references [37, 38] with permissions. AE, auto-encoder; CARS, coherent anti-stokes Raman scattering; SHG, second-harmonic generation; TPEF, two-photon excited fluorescence microscopy

a differentiable function represented by an MLP (or an AE) that maps this input to an output $y_{\mathcal{G}}$, such that $\mathcal{G}(\mathbf{x};\Theta_{\mathcal{G}}):\{\mathbf{X},\mathbf{z}\}\rightarrow y_{\mathcal{G}}$. The generator $\mathcal{G}(\mathbf{x};\Theta_{\mathcal{G}})$ aims to learn the distribution $p_{\mathcal{G}}$ to approximate the prior distribution of the real data p_{data} from where the input $\mathbf{X}$ was drawn. The output $y_{\mathcal{G}}$ of the generator has visual similarity with the real data, e.g. images. In addition to the output from the generator, a real input image is also fed to the discriminator $\mathcal{D}$. The output of the discriminator $\mathcal{D}\left(y_{\mathcal{G}};\Theta_{\mathcal{D}}\right):y_{\mathcal{D}}\rightarrow[0,1]$ represents a probability that $y_{\mathcal{G}}$ is retrieved from p_{data} rather than $p_{\mathcal{G}}$ (see Figure 8). Both the networks $\mathcal{G}$ and $\mathcal{D}$ follow a min-max game where $\mathcal{D}$ minimizes the probability of $y_{\mathcal{G}}$ belonging to p_{data}, and simultaneously $\mathcal{G}$ maximizes this probability by generating more realistic images that cannot be distinguished by $\mathcal{D}$. This adversarial training is achieved by optimizing the loss function

$$E(\mathcal{G},\mathcal{D})=\mathbb{E}_{\mathbf{x},y_{\mathcal{G}}}\left[log\mathcal{D}\left(\mathbf{x},y_{\mathcal{G}}\right)\right]+\mathbb{E}_{\mathbf{x},\mathbf{z}}[log(1-\mathcal{D}(\mathbf{x},\mathcal{G}(\mathbf{x},\mathbf{z})))] \tag{9}$$

with back-propagation technique. During back-propagation, the gradient calculated over the loss function is back-propagated from the discriminator to the generator, in order to update the parameters of the generator. While training a GAN network certain challenges are encountered. Foremost, it is difficult to obtain convergence of both the networks due to simultaneous training of the networks. Additionally, an early convergence of the discriminator network can cause the generated images to be easily distinguished from the true images. This is a consequence of the gradient of the discriminator reaching zero and thus providing no guidance to the generator for further training. After a few iterations, when convergence between the two networks is achieved, ($p_{\mathcal{G}}=p_{\text{data}}$ and $\mathcal{D}(\mathbf{x})=\frac{1}{2}$) the generator can produce realistic images, which are difficult to identify as "fake" images [41] by the discriminator.

Such an adversarial training of GANs have gained popularity in industrial and academic research due to their capability of domain adaptation and generating new images. Generative adversarial networks (GANs) are potentially used for biophotonic applications including denoising of images, correcting stitching artifacts in microscopic images, increasing spatial resolution [42, 43], virtual H&E staining of fluorescence images [44] and biological image synthesis of fluorescence images [45, 46] (see Figure 9). The applications of GANs are elaborated in chapter 3.

All the above mentioned deep learning architectures are huge and have many layers. With increasing architecture and dataset size, the memory requirements increases as well. Therefore, high computational power and efficient software are needed. A detailed explanation of the hardware requirements and commonly used software is given in the section 2.6.

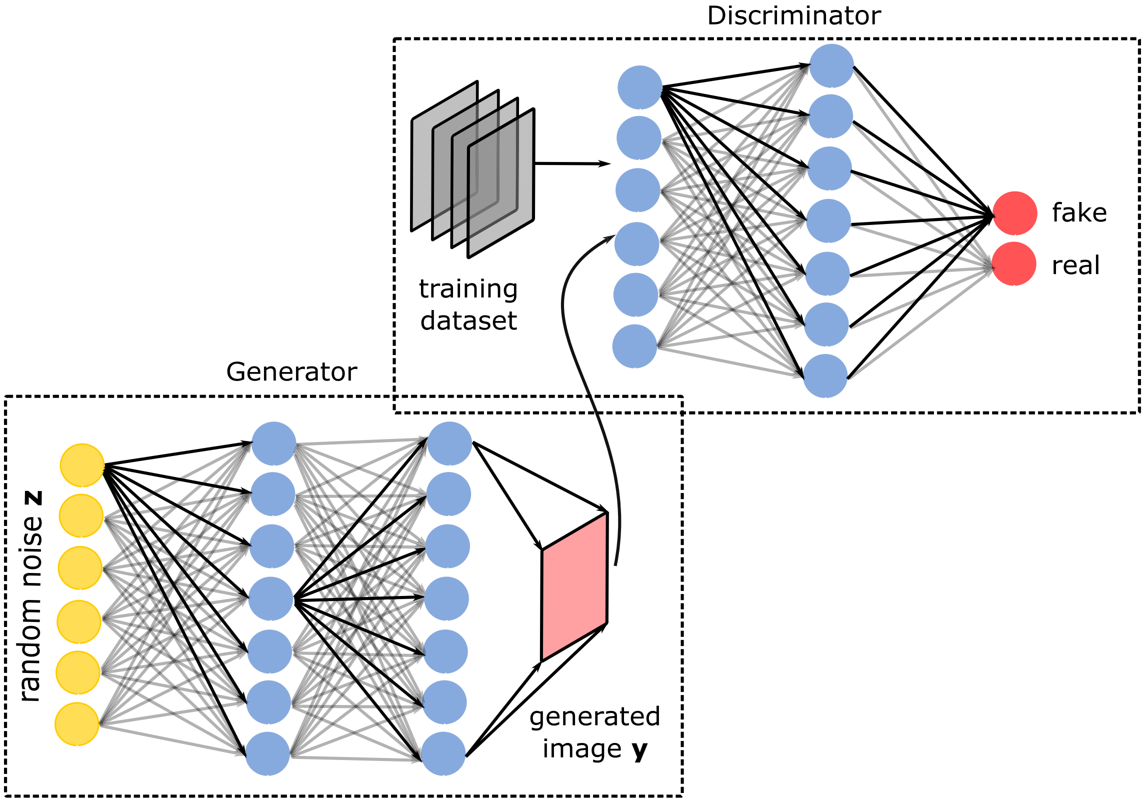


FIGURE 8 Generative adversarial network shows two adversaries, a generator and a discriminator. A generator's input is either random noise **z** or an image **X**. The output from a generator y_G, is fed to the discriminator $\mathcal{D}$ which distinguishes the generated output as real or fake. Both the networks are adversaries of each other as both the networks optimize different objective functions

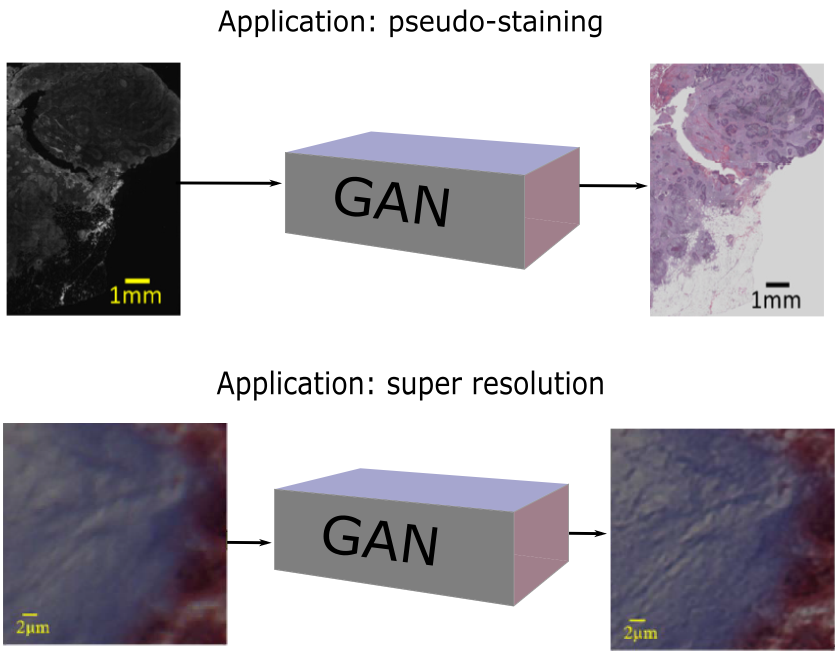


FIGURE 9 The two common applications of GANs including pseudostaining (upper panel) and resolution enhancing (lower panel) are shown. The image in the upper panel utilizes an autofluorescence image as an input and the GAN network produces H&E stained image at the output. Similarly, the image in the lower panel shows that a GAN model was used to enhance the resolution of a Masson's trichrome stained lung tissue section. The images are reprinted from earlier researches [42, 44] with permission. GAN, generative adversarial network

2.6 | Hardware throughput and software libraries

Deep learning algorithms perform complex matrix multiplications of millions of parameters in the hidden layers. This limits the performance of deep learning models due to the need for higher computational power and memory size. The recently introduced GPUs provide higher computational power as compared to conventional CPUs, thereby, accelerating the training of deep learning models to a greater extent.

In addition to the hardware, the availability of various software packages can facilitate the use of deep learning models in biophotonics. A range of open-source deep learning libraries like Caffe [47], Torch [48], Theano [49], Tensorflow [50], Keras [51] and Lasagne [52] are developed along with their interfaces in C++, Python and Lua programming languages. These packages can be efficiently implemented with GPUs, thus accelerating the training of deep learning models. Various researches using these libraries have been conducted for spectroscopic data which is discussed in section 3 and section 4.

2.7 | Educational resources

The above sections provide brief information about deep learning and various architectures. However, to make deep learning algorithms profitable for the biophotonic community, various educational resources are mentioned in this section.

In this context, books by Nielsen [53], Ripley [7], Russel and Norvig [8], Bishop [54], Goodfellow [6] and many more [55–58] are recommended sources for deep learning. Additionally, there are several online courses for deep learning which can give an overall hands-on experience of using various deep learning models with Python and R programming languages. Some of them are listed here: Coursera (https://www.coursera.org), deeplearning.ai (https://www.deeplearning.ai) and others (https://www.datacamp.com, https://machinelearningmastery.com, https://www.pyimagesearch.com).

Furthermore, applications of deep learning are showcased at a number of international conferences dedicated to biophotonics. A few of them include, but are not limited to, SPIE conferences, OSA conferences, IEEE conferences and FACCS conferences. Likewise, many peer-reviewed journals fully dedicated to the field of biophotonics have embraced the applications of deep learning and attracted interdisciplinary readership.

In the next two sections, applications of deep learning are elaborated.

3 | DEEP LEARNING FOR BIOPHOTONIC IMAGING

In the past decade, biomedical optical imaging has witnessed a vast development ranging from fast scanning systems to automated image analysis algorithms. In addition, developments like increased penetration depth, molecular specificity, faster image acquisition and high spatial resolution are advantageous for bed-side patient monitoring and diagnostics for personalized treatments. However, due to practical limitations of optical systems, certain challenges are encountered with the fast acquisition of highly resolved and noise-free data. Recently, deep learning algorithms have been used to address these unmet needs in biophotonic imaging and has shown overwhelming results for a broad range of applications. These applications will be further discussed in this section.

3.1 | Image denoising/deblurring

Deep neural networks can be designed for virtually any kind of input–output combination. One way to employ deep neural networks is to feed noisy or low-resolution images to the input of a generative network and use the images with desired resolution or noise level as an output. The generative network, which learns features from the high-resolution images, can be subsequently used for image enhancement. Generative networks using mean square error or similar type of loss function often lead to overly smoothed images at the output. A common way to preserve high-frequency features is to build a generative adversarial network (GAN), which was described in more details in section 2.5. Shortly, the GAN network contains a generative network to produce an image and a discriminator network to estimate the quality of the image produced by the generator. A variation of this architecture, called Wasserstein generative adversarial network (WGAN), which uses Wasserstein distance as a loss function, was recently utilized for resolution enhancement of OCT images [59]. An alternative, edge-sensitive conditional generative adversarial network (cGAN) was reported efficient against speckle noise. This speckle noise reduction was demonstrated for OCT images which utilized an edge-sensitive cGAN [60]. Another implementation of the GAN approach with additional content loss metrics was proposed for simultaneous denoising and super-resolution generation of optical coherence tomography [61]. This content loss was calculated from the difference between features extracted from the true image and the generated image. Besides OCT images, the GAN approach was successfully applied for fluorescence microscopic images, making a cross-modality super-resolution possible without employing overly sophisticated setups [43]. The approach of achieving super-resolution by deep learning is additionally discussed in section 3.7.

In all above examples, deep neural networks learned patterns from the data, which makes it possible to increase the resolution and the signal-to-noise ratio simultaneously. This makes these methods advantageous in comparison with classical image enhancement methods, which usually improve one of the two quality parameters at the expense of the other parameter.

3.2 | Semantic segmentation

Semantic segmentation is a pixel classification task, where every pixel of an image represents a class. Semantic segmentation is widely used in digital pathology for applications, like tissue segmentation, nuclei segmentation and lesion detection [62]. Similarly, semantic segmentation of microscopic images, like nonlinear multimodal images [37, 63], OCT images [64] and fluorescence images, using auto-encoders (see section 2.4) is gathering researcher's interest. The above-mentioned works utilize U-net [65] type networks, which is an auto-encoder architecture with special connections between the encoder and the decoder network. Another striking feature of U-net is the weighted loss function, which heavily penalizes the misclassification of boundary pixels of an object, thus allowing to segment closely located objects efficiently. Previous research showed the semantic segmentation of nonlinear multimodal images (CARS, TPEF, SHG) of lung tissue using the U-net [63] architecture and of gastrointestinal tissue regions using the SegNet architecture (see Figure 7 bottom) [37, 66], respectively. Similarly, the authors of a recent research article [38] segmented a Drosophila heart in optical computed tomography images based on an U-net architecture (see Figure 7 top). Furthermore, it is shown in a recent work [37] that CNN based semantic segmentation achieved better performances as compared to traditional machine learning methods.

In addition to CNNs, recurrent neural networks have also shown promising results for semantic segmentation of the CamVid dataset [67, 68]. A recent work [69] used RNNs for perimysium segmentation in H&E stained skeletal microscopic images and achieved better performance as compared to the U-net architecture. RNNs can retrieve global spatial information of an image, which improves the semantic segmentation performance [69]. However, training a RNN can be computationally expensive and therefore it is underexplored in biophotonics.

3.3 | Disease recognition

Disease recognition using MLPs and CNNs is a very common application in the field of biophotonics. Out of all deep learning architectures discussed in section 2, MLPs are widely used for disease recognition and assessment. For example, MLPs were used to classify FLIM data of cervical neoplasmic tissue sections, which achieved a significant discrimination between the normal and the precancerous group as well as between the low-risk group and the high-risk group [70]. Another application of MLPs was reported using Raman spectroscopic data for classification of patients with Alzheimer's disease, other types of dementia and healthy individuals. Comparison of MLP results with conventional classifiers, like the radial basis function (RBF) classifier, showed that MLPs outperformed the conventional classifiers for the tested classification tasks [71]. In addition to MLPs, CNNs are the second most widely used deep learning architectures for disease classification. A recently proposed CNN application [72], classified malaria infected blood smear from healthy controls using Leishman stained images. The malaria-infected images were further used to segment the infected RBCs. Similarly, a very recent research reported the use of CNNs to assign cervical cancer into three stages using CARS, SHG, TPEF microscopic data [73].

Mostly, the data acquired by spectroscopic techniques is small, due to larger acquisition times. Therefore, the training of MLPs or CNNs is always challenging due to the small datasets available. In such cases, deep learning networks using transfer learning strategies can be applied [26, 74–76]. Transfer learning utilizes CNN models pretrained on a (large) source dataset and transfer the learned features to classify a (small) target dataset. For example, pretrained CNN models including GoogleNet [77], Inceptionv3 [78] and VGG16 [79] were used to classify breast cancer in OCT images [75], head and neck cancer in 3-D OCT images [80], lung cancer in CARS images [74] and hepatocellular carcinoma in multiphoton microscopic images [26] (see Figure 4 top), respectively. Here, the CNN models are first trained on large nonbiological datasets like ImageNet [81] and the parameters of these pretrained models are fine-tuned on the new biophotonic dataset.

To summarize, MLPs and CNNs are majorly used for disease classification. Recently reported transfer learning strategies [26, 74, 75] using CNNs have worked best for small datasets, which are accounted in biophotonic studies.

3.4 | Cell or organ localization

In addition to the segmentation tasks described in section 3.2, there is a specific segmentation application known as the "localization" task. In biomedical imaging, localization can be used for counting cells of a specific type within the sample or its image. Subsequently, the localized cells can be segmented and analyzed through descriptive statistics over cell sizes, shapes and the cell morphology. Alternatively, segmented cells can be automatically classified or investigated manually by pathologists. It was shown that leukocytes can be localized within blood smear images and segmented using deep neural networks efficiently [27]. For the leukocyte localization, a multistep workflow that included a feature extraction by a feature pyramid network inspired by the ResNet architecture [82] was utilized. This was followed by the determination of a region of interest. Thereafter, a localization box was predicted and the leukocytes were segmented (see Figure 4 bottom). On every step of this workflow, convolutional or fully connected layers were used instead of user-defined features.

Another biomedical application of deep learning is organ localization within 3D computed tomography (CT) scans, which is an essential preprocessing step for the analysis of the scans. Recently, the organ localization and segmentation within 3D scans was demonstrated using a 3D U-net approach [83] and a 2D multichannel SegNet model [84].

3.5 | Pseudostaining

In imaging of biological tissue and cell samples often histological staining needs to be applied in order to enhance the contrast and highlight tissue features. This staining is usually performed during the sample preparation prior to the microscopic investigation of the sample. Both manual and automated microscopic image analysis often require such stained images. Some stains, like the hematoxylin and eosin (H&E) stain are used over many decades as "gold-standard" techniques in pathology. The main drawback of the conventional staining techniques is that they require additional time and effort. Recent studies showed that in certain cases deep learning can be employed instead of the actual sample staining. It was shown that cGAN architecture can be used to generate H&E stained images from hyperspectral microscopic images of unstained samples [85]. Another study employed a CNN-GAN approach in order to obtain H&E stained images from unlabeled tissue autofluorescence images (see Figure 9 top) [44]. Both studies performed virtual H&E staining by using different imaging techniques in a combination with deep learning instead of actual staining the sample. On the other side, it was shown that deep learning makes it possible to restain H&E stained microscopic images into immunohistochemical (IHC) stained images [86]. The advantage of such approach is that H&E is a conventional and simple staining but the IHC staining is more costly and labor intensive. For such restaining, a conditional CycleGAN (cCGAN) architecture was used. Oversimplified, this CycleGAN approach is a combination of two generators (encoder and decoder) and discriminators. The first generator produces an IHC stained image from a H&E stained image, subsequently, the second generator transforms the generated IHC image into a virtually stained H&E image. This cycle makes it possible to introduce cycle identity loss and classification cycle loss in the network architecture.

In prospect, deep learning in combination with various imaging techniques may provide a fast and flexible alternative to the histological staining, making it possible to switch between virtual stains without additional sample preparation and measurements.

3.6 | Image registration

Nowadays, it is a common practice to measure one sample with multiple modalities in order to achieve a comprehensive characterization of the biological tissue specimen. For the joint analysis of the images obtained from two or more modalities, a perfect overlay of the two images is required. This is termed as image registration. The basic idea of the image registration methodology is to minimize or maximize an objective or a cost function computed on the overlapping region of the two (moving and fixed) images. The optimization of the objective function is achieved by iteratively searching a geometric transformation for the moving image. Various semiautomatic approaches have been proposed by researchers to register secondary ion mass spectrometry images with optical image [87], Raman microscopic images with mass spectrometric MALDI-TOF images [88] and FTIR images of tissue microarray (TMA) cores against H&E images [89]. However, these methods are not fully automated and require manual intervention. A recently developed automatic approach based on a sparse search strategy deals with sub region registration of FTIR microscopic images in whole-slide histopathological staining images. Additionally, the FTIR imaged cores of tissue microarrays were registered with their histopathologically stained counterparts. This work also presented the registration of CARS images within histopathological staining images [90]. Although this approach is robust and reliable for diverse microscopic technologies, it requires preprocessing of the samples acquired from various modalities. In such cases, CNN based registration can potentially register images obtained from different modalities without the need of image preprocessing.

Recently, CNN based registration methods have been reported in radiology, which learn geometric transformation parameters for registering MRI and CT images [91, 92]. The results from the CNN based registration have shown surprisingly good results and are efficiently applied in a multiresolution scenario. However, a CNN based image registration of spectroscopic images is still under-explored and requires further investigation.

3.7 | Image super-resolution

An earlier section (see section 3.1) discussed that GAN architectures can be employed for both image denoising and resolution enhancement. Although the improvement of the signal-to-noise ratio is very important for the interpretation of the images, it can also be improved by increasing the number of collected images. On the other side, the resolution of the obtained image is often limited due to the technical properties, like the diffraction limit. There are various sophisticated technical solutions, which allow an imaging below the diffraction limit. A class of such techniques is called super-resolution imaging.

Besides technical solutions, overcoming the diffraction limit is also possible by employing image processing techniques, and in particular, deep learning. Studies showed that CNN can be applied to effectively improve the resolution of the stained tissue section (see Figure 9 bottom) [42]. A fully convolutional encoder-decoder network was successfully constructed for imaging of quantum dots and microtubes using single-molecule localization microscopy [93]. Another imaging limitation was pushed by deep learning in the area of lens-free holographic microscopy (LFHM). Due to the absence of the lens, the resolution is limited by the pixel size of the detector. To overcome this issue a CNN network, inspired by an U-net architecture was employed for LFHM, which made it possible to perform pixel super-resolution imaging [94]. Another example of generating super-resolution images was implemented for OCT images using a GAN-based approach [61]. Besides achieving super-resolution, this GAN-based approach decreased the image noise simultaneously.

In addition to the above-mentioned applications, deep learning is vastly applied for vibrational spectroscopic data including applications like preprocessing and classification of spectra. These applications are discussed in the following section.

4 | DEEP LEARNING FOR VIBRATIONAL SPECTROSCOPY

Until recently, data analysis in vibrational spectroscopy employed well-established classical machine learning techniques adapted to the structures of specific spectroscopic data. The general workflow in these scenarios is composed of preprocessing, feature extraction or feature selection and statistical modeling [95]. In contrast to the widespread use of artificial neural networks in spectral analysis [96–98], the application of deep learning in this field is growing but still in the early stage. This is because, on the one hand, classical machine learning does a great job in most cases, and on the other hand, deep learning in spectral analysis encounters many difficulties. Most of the existent deep neural networks were developed for image analysis or speech recognition and cannot be directly transferred to spectral analysis. Building a deep neural network for spectral analysis from scratch requires a lot of hyperparameter tuning and is tedious. Unlike in image analysis, there is rarely a pre-trained deep learning model for spectral data. The lack of large spectral datasets forms another difficulty to apply deep learning in spectral analysis. Nevertheless, the spectral analysis does see benefits from deep learning, which will be discussed in the following section from the perspectives of spectral preprocessing and statistical analysis.

4.1 | Preprocessing

Spectral preprocessing aims to remove corrupting contributions from the measured spectra, which is often done by smoothing, baseline correction, standardization, and so on. Preprocessing is a burden, not only because of the computation time, but also because it is not straightforward to select the preprocessing techniques that perform best on each specific dataset [99]. Deep learning can be a time saver assuming that the deep neural network is powerful enough to tolerate the corrupting effects and can be trained on raw data without any preprocessing to reach a satisfying performance. This has been shown in references utilizing convolutional neural networks or stacked contractive auto-encoders [100–104]. The kernels of the trained network were shown to work as smoothing, derivative/slope recognizers, thresholding and spectral region selection, which are basically preprocessing steps [101]. Unlike conventional preprocessing approaches, however, the outputs of the kernels are not necessarily physically meaningful, but rather a mathematical representation of preprocessing for the given data. This representation is best suited for the following regression or classification models. Nevertheless, a close inspection of the outputs of the kernels does give a hint about the features that are the most significant for the regression or classification [101, 103].

While most investigations are engaged to construct deep learning methods utilizing the raw data and skip preprocessing, there are indeed efforts to apply deep learning as a preprocessing approach, especially for issues that cannot be solved easily with conventional preprocessing methods. As it is widely known, a sufficiently long integration time is normally needed for a usable spectrum, especially for Raman spectroscopy considering the small Raman cross-section. The slow measurement, especially in the case of Raman imaging, has hindered Raman spectroscopy to be applied for the investigations of dynamic processes. In such cases, fast measurements are needed but they suffer from bad data quality, such as extremely high noise or low spectral/spatial resolution. Deep learning has shown its capability of handling this issue in recent publications [105, 106]. For example, an U-net was applied to stimulated Raman spectra to reduce noise in the data and hence improve the sensitivity, which helps shorten the spectral acquisition time down to 20 μs without losing sensitivity [105]. In another investigation [106] the authors applied a deep convolutional neural network to improve the spatial resolution of the Raman hyperspectral data. In this way, the line-scan Raman measurement was largely accelerated.

Following the spectral preprocessing, investigating the spectral data by using multivariate statistics and classification models is commonly performed. The next section discusses the statistical modeling of spectral data using deep neural networks.

4.2 | Statistical modeling

It is commonly hypothesized that deep neural networks are capable of feature learning [107], that is, they do not require hand-engineered features, which are needed to apply conventional classifiers. With multiple layers of linear and/or nonlinear units, deep neural networks show huge potential to learn hierarchical representations of features from complex data. It is thus advantageous to apply deep neural networks for the analysis of vibrational spectra, which are a complex superposition of all vibrational information within the sample. Applications of deep learning were reported for both infrared and Raman spectroscopy in order to achieve tasks like brain function investigations [108, 109], biological diagnostics [102, 110, 111], cytopathology [112], microbial identification [113], pathogenic bacteria identification [113], food science investigations [114, 115], tobacco leaves characterization [116] and mineral analysis [117]. Furthermore, it was reported in references that deep learning can perform better than classical machine learning methods [100, 103]. A deep convolutional neural network was also used for an un-mixing tasks, i.e., to resolve pure components and their abundances from mixture spectra. Thereby, *N* one-component identification models were trained with data composed of spectra of a pure component, negative and positive samples in terms of this pure component. The *N* models could successfully solve the unmixing task at the end [118].

In addition to the different applications discussed above, strategies were reported to improve the performance of deep learning. In particular, a hierarchical deep convolutional neural network was employed on Raman microscopic data, in which neighboring spectral pixels were merged hierarchically in order to combine the spatial information with spectral information. This combination finally led to a better classification between healthy and cancer cells [112]. In addition, different searching algorithms such as grid search [103], particle swarm optimization (PSO) [114] and artificial bee colony algorithm (ABC) [117] have been utilized to automatically find the optimal hyperparameters of a deep neural network. A combination of a CNN and an extreme learning machine (ELM) was reported to speed up the training and improve the generalization performance of the trained network. The optimal values of ELM were sought by an artificial bee colony algorithm (ABC) [117].

Despite the investigations included in previous paragraphs, deep learning is far less developed in vibrational spectral analysis in comparison with image analysis and speech recognition. One of the reasons is that the deep neural networks are extremely data starving, but measuring spectral data from a large number of samples is limited by practical reasons, especially for biological samples. Data augmentation can be utilized to solve this issue, which is normally done by randomly shifting the wavenumber axis, adding random noise and/or (linearly) combining multiple spectra [100, 101]. However, these data augmentation techniques can introduce unknown (spectral) features into the data, especially if the variations of interest are very subtle. This is perhaps the reason, why the best model achieved in reference [101] was trained by utilizing an additional EMSC after data augmentation. A generative adversarial network may play a role for better data augmentation, but there is yet no application reported to the authors' best knowledge.

Besides the intrinsic complexity of the spectra and limited sample size, vibrational spectroscopy is remarkably sensitive to measurement conditions and there exist significant variations among multiple measurements. Hence, it is important during spectral analysis to learn features of interest but not those related to the measurement in order to achieve an optimal prediction on new measurements. Deep learning can play a role in this context as it was reported in previous research [101]. Therein a CNN was used to predict a test dataset comprising of drug concentrations higher than the concentrations of the training dataset. In this case, the test performance of the CNN can be improved, only if the hyperparameters of the network was tuned based on the validation set. Tuning with a randomly selected validation set did not provide significantly better predictions. In fact, it is more than difficult to build a deep neural network, which tolerates unwanted variations and generalizing well between measurements. Data augmentation can help in this situation, as it was discussed in reference [101], but the improvement was limited. Another strategy is transfer learning, which has been discussed in the previous section. Its capability for dealing with unwanted spectral variations was shown in reference [102] where the deep network was pretrained on embedded tissues and finetuned to classify fresh frozen tissues.

Another important issue of applying deep learning for vibrational spectral analysis is a proper validation. As it was mentioned in the last paragraph, vibrational spectra often vary from measurement to measurement and device to device. It is thus important and necessary to validate a deep neural network using measurements independent to the training data. A random separation between training and testing data should be avoided. In addition, the testing data cannot be included in any procedure that affects the final modeling, including model-based preprocessing such as EMSC [119]. Otherwise, an overestimation of the network is highly possible. Similar challenges and issues related to deep learning methods are discussed in the next section.

5 | DISCUSSIONS AND CRITICAL ISSUES

Deep learning was already applied several times in biophotonic data analysis, but its potential is much larger. To use this potential an immense amount of data for training is needed. If such large datasets are not available, then increasing the dataset size by data augmentation or using transfer learning methods to achieve good model performances are commonly used approaches. Furthermore, class imbalances are predominantly seen in clinical studies, which affect the training of deep neural networks. Another issue about using deep learning methods is the lack of interpretability of model predictions, which restricts the use of deep learning methods for newly developed measurement modalities in the biophotonic field. Additionally, proper model validation techniques are needed, which will be elaborated in this section.

5.1 | Current challenges

This subsection elaborates the challenges which are related to the dataset, training and understanding of the deep neural network encountered by data scientists in biophotonics.

5.1.1 | Lack of data

Biophotonic technologies are emerging techniques with restricted use in clinical practice as compared to other radiological and conventional histopathological techniques. Therefore, the dataset size is often limited. Moreover, the systematic accessibility of data and open repositories is limited in the biophotonics field. This leads to one of the major challenges to use deep learning for biophotonic data, which is the shortage of data. Deep learning models are data-driven and require a large amount of data depending on the task and the number of parameters in the model [120, 121] (Table 2).

Small datasets can easily lead to over-fitting causing poor generalizability on a new dataset. The problem of small datasets can be overcome by increasing datasets using data augmentation techniques. The basic idea of data augmentation is to artificially expand the training dataset by creating modified versions of the original dataset. For example, commonly used data augmentation techniques for image data are translation, rotation, shifting, increasing or decreasing brightness and magnification of the images. Other commonly used data augmentation techniques for images are adding Gaussian noise and transforming the color space of the images [122]. Likewise, data augmentation of spectral data can also be performed by adding noise to the spectral data or shifting the wavenumber axis for spectroscopic data [100, 101]. However, it is worth noting that slight perturbations in the images or the spectra can also degrade the model performance [123]. To prevent the degradation of the model performance and also to avoid too large dataset sizes, we discuss some practical considerations with the perspective of data augmentation in section 5.2.1.

In addition to data augmentation, transfer learning is another alternative technique to train deep learning models on small datasets. This technique focuses on transferring features of a deep neural network learned on a larger dataset to a small dataset. Research has shown that transfer-learning strategies lead to promising results

TABLE 2 Abbreviation in alphabetical order

Acronym	Explanation
ABC	Artificial bee colony algorithm
AE	Auto-encoders
ANN	Artificial neural network
BRNN	Bidirectional recurrent neural network
CARS	Coherent anti-stokes Raman scattering
cGAN	Conditional generative adversarial network
cCGAN	Conditional cycle GAN
CGAN	Cycle GAN
CNN	Convolutional neural network
CPU	Central processing unit
DBN	Deep belief network
ELM	Extreme learning machine
EMSC	Extended multiplicative signal correction
GAN	Generative adversarial network
GPU	Graphical processing unit
GRU	Gated recurrent unit
H&E	Hematoxylin and eosin
IHC	Immunohistochemical
FLIM	Fluorescence life-time imaging
FTIR	Fourier-transform infrared spectroscopy
LFHM	Lens-free holographic microscopy
LSTM	Long short-term memory
MALDI-TOF	Matrix assisted laser desorption-ionization (time of flight)
MLP	Multilayer perceptron
MPM	Multiphoton microscopy
OCT	Optical coherence tomography
PAI	Photoacoustic imaging
PSO	Particle swarm optimization
RBF	Radial basis function
RBM	Restricted Boltzmann machine
ReLU	Rectified linear unit
RNN	Recurrent neural network
SAE	Stacked auto-encoder
SERS	Surface enhanced Raman spectroscopy
SGD	Stochastic gradient descent
SHG	Second harmonic generation
SMOTE	Synthetic minority over-sampling technique
TPEF	Two-photon excitation fluorescence
WGAN	Wasserstein generative adversarial network

when applied for small spectroscopic dataset [26, 74, 75]. However, transferring features of a deep neural network which is pretrained on a dataset like ImageNet, to perform classification or regression tasks on spectroscopic data, is debatable. Prior research has shown that with increase in the distance of the tasks (like classification, regression) and domains (like biological, non-biological), the transfer of the specific features learned in the last layers of a deep neural network can negatively affect the model performance. Thus, leading to “negative transfer learning” [124]. A practical advice on applying transfer learning approaches on small dataset is given in section 5.2.2.

5.1.2 | Imbalanced dataset

A second challenge in training deep learning models is an imbalanced data distribution, which is a key issue in all biological datasets. Training a deep neural network with unbalanced datasets affects the sensitivity of the loss function towards the majority class. To circumvent such biases, data-level and method-level approaches are used. Data-level methods address the class imbalance problem by random over-sampling the minority class or under-sampling the majority class. Although data-level methods are simple, over-sampling can introduce over-fitting of the model and under-sampling can cause loss of important information. Another complex sampling method is synthetic minority over-sampling technique (SMOTE), which creates synthetic data for the minority class. However, this method is limited due to the issue of generalizability and variance [125]. Also, creating synthetic spectral data is not straight forward due to the complexity of the spectral features.

An alternative to this imbalance issue are model-level methods, which have significantly improved the training results of deep learning models. In these cases, the loss function is penalized by the weight of the classes, which is defined by the number of samples in each class. However, sometimes it is difficult to define a customized loss function for a multiclass classification task. Many researchers have reported the use of a hybrid approach, where data-level and model-level methods are combined. Furthermore, other methods dealing with the loss function to overcome class imbalances have also been reported in the literature [126, 127].

5.1.3 | Bias-variance trade off

The third challenge encountered while constructing any machine learning method is the bias-variance trade off. There is always a competition to find a perfect balance between high bias (under-fitting) and high variance (over-fitting) for complex models. Model complexity can be defined as the number of trainable parameters in a model and an increase in number of trainable parameters also increases the model complexity. With increasing model complexity, like encountered in deep neural networks, the increase of variance is more likely. A high variance in deep learning models can be due to three major reasons: the first reason is the sampling variance, the second reason is the model complexity variance and lastly is the model initialization variance. Sampling variance is a consequence of a high biological variance between the samples and within a sample (eg, the variance between biological replicates and within the replicates). Therefore, acquiring more balanced data and maintaining a consistent data acquisition protocol is essential. Additionally, comparison of the data acquired in different laboratories and devices should be encouraged in order to avoid such biases.

Model complexity and model initialization variance is controlled by the depth and width of the deep neural networks. Research has shown that an increasing the depth of a deep neural network by adding layers to a neural network can be a source of over-fitting, whereas increasing the width of the deep neural network decreases the model-related variance [128]. Therefore, designing a deep neural network should be done with focus on the generalization capabilities of the model. Even though, bias-variance trade-off is also observed in classical machine learning models, research shows that deep learning methods can efficiently find a balance between the bias and variance [129, 130].

5.1.4 | Interpretability of the "black-box"

Deep learning models have achieved breakthrough performance in various domains of medical imaging including biophotonics (see section 3 and section 4). As these models are intended to be utilized in modern healthcare systems, the interpretation of their decision-making is a key issue. It is important to know if deep neural networks make their predictions based on the biomolecular information instead of some background effect or noise in the spectroscopic data. An example of missing interpretability of the "black-box" models can be seen in a recent research [37] where an auto-encoder like model was used to segment nonlinear multimodal images of CARS, TPEF and SHG into four tissue regions. The segmentation results from the auto-encoder were satisfactory compared to the classical machine learning approach using hand-engineered texture features. However, the contribution of the three modalities CARS, TPEF and SHG for the segmentation of crypts was unknown. Similarly, by using deep learning models the contributions of spectral features to a prediction, like the presence or absence of a disease, is difficult to interpret. This drawback hinders the usage of deep learning models especially in newly developed biophotonic technologies. Nevertheless, researchers are now developing various decomposition techniques for understanding complex deep learning models [131–135].

A recent research [136] utilizes Taylor series expansion for interpreting the output function of nonlinear models like ANNs on Raman spectroscopic data. Within this approach, the degree of nonlinearity of ANN model was realized using a second-order Taylor expansion. This allowed an interpretation of the patterns learned by ANN models based on wavenumber combinations to predict a particular class. Another approach [131] uses the layer-wise decomposition of features from hidden layers to understand the contribution of all pixels in an image to detect a particular class. While all these techniques are mostly developed for computer vision tasks, its utility can be expanded for spectroscopic data and this needs further investigations.

5.1.5 | Standardization for biophotonics

Biophotonics has an outstanding potential for clinical healthcare. However, in contrast to the well-established radiological or histopathological techniques, biophotonic technologies lack the adoption of standard procedures. There are no international consensus of assessing the performance of biophotonic devices which largely affects the reproducibility of data. Subsequently, the machine learning models trained on such data are less reliable. In this regard, several publications [137–139] have presented standardization procedures for various biophotonic technologies.

Improving the quality of clinical studies, comparing data from different laboratories and systems, facilitating the use of open databases, allowing quantitative comparisons between different models are critical factors for developing the best computational models. Validating the strength of these machine-learning models is also important and is further discussed in section 5.2.4.

5.2 | Practical considerations: *do's* and *dont's*

Researchers often encounter challenges as it was discussed in section 5.1 while training a deep learning model. To overcome these challenges, various approaches including data augmentation, transfer learning and model validation are established. However, these approaches have pitfalls that can generate poor deep learning models, increase the training time and cause memory issues. Thus, it is important that developers circumvent common pitfalls while constructing deep learning models. In the following section a practical advice for constructing these models and avoiding common mistakes are given.

5.2.1 | Data augmentation

The choice of data augmentation should be made depending on the dataset. Data augmentation strategies like horizontal flips, random rotations, scaling and shearing are simple to implement, however these strategies fail to add new information or patterns into the training dataset [140, 141]. Moreover, random rotations and translation can introduce zero values in the corners of the image, which causes a bias in training the deep neural network. Therefore, the image regions with zero values are removed or filled with a reflection of the original image. In addition to geometric transformations, adding noise like jitter or Gaussian noise has improved regularization properties of the deep neural network for medical image classification [140, 142]. For fluorescence images, Gaussian and Poisson noise are commonly observed. These can be simulated to generate synthetic fluorescence images. Another data augmentation technique is the style transformation using GANs, commonly known as style transfer. In style transfer methods, the color and texture information from one image is transferred to another image to generate a completely new image [141, 143]. However, style transfer in biophotonics requires systematic investigation, as it may cause subtle alterations in the color and texture of the newly generated image which are associated to the biomolecular information under investigation. Thus, data augmentation techniques like style transfer should be performed cautiously for medical imaging, because it may also require changing the labels respectively. Another method to create large datasets from small dataset is the extraction of patches of the images. This method was implemented in a recent research [37] for semantic segmentation of nonlinear multimodal images. Utilizing patches for data augmentation not only increases the dataset size but also retains the biomolecular information of the images without the need to change labels. However, extracting patches of large spectroscopic images fails to generate new independent data and contrarily increases the dataset size. This can cause memory requirement issues. In such cases, large images should be down sampled and non-informative patches should be removed.

As mentioned above, data augmentation can increase the dataset size and memory requirement depending on the data augmentation scheme applied. To tackle this issue, online and offline data augmentation strategies can be chosen. If the dataset is relatively small, offline data augmentation can be performed. Offline data augmentation increases the dataset size by a factor equal to the number of transformations performed. If the whole augmented dataset is used for model construction, it can increase the memory requirements. The second option is online data augmentation which performs transformations of the mini-batches used while training the deep neural network model. This approach reduces the memory requirements but increases the training time.

In addition to the above-mentioned points, there are further important considerations for data augmentation. First, data augmentation should be performed for the training dataset only. Moreover, all the images should be rescaled to the same size before adding any kind of noise and various levels of noise can be tested to achieve the best validation accuracy. Overall, the benefit of data augmentation in biophotonics is an open issue that should be investigated systematically.

5.2.2 | Transfer learning

The previous section explains that data augmentation is an effective method to work with small datasets and this section introduces transfer learning as a strategy for small datasets. There are two transfer learning strategies which are commonly followed: first, a pretrained deep neural network are used as feature extractor and those features are utilized to build an easy model for classification or regression. The second strategy is to fine-tune the weights of a pretrained deep neural network using the new dataset. Fine-tuning of the weights can be conducted for all the layers of the network or restricted only to the last layers where most specific features are learned. Based on the two transfer learning strategies, the size of the dataset, the similarity between the datasets and the similarity between the tasks (classification or regression) involved, four major approaches can be utilized [144]:

- If the new dataset is small and similar to the original dataset, then the generic features from the top layers of a pretrained deep neural network will be relevant for the new dataset and thus these generic features can be used to train an easy classifier.
- If the new dataset is large and similar to the original dataset, then fine-tuning of the whole pretrained deep neural network can be performed.

- If the new dataset is small and different from the original dataset, then it is best to train a linear classifier (linear discriminant analysis or support vector machine) by using activations from the top and intermediate layers of a pretrained deep neural network. Previous research reported that this method works best for small spectroscopic datasets [26, 74, 75]. However, for biophotonics this needs proper investigation depending on the dataset.
- If the new dataset is large and different from the original dataset, then it is beneficial to train a deep neural network from scratch and initialize the weights using a similar pretrained deep neural network model.

5.2.3 | Splitting the dataset

Splitting of the dataset depends on the dataset size. In many machine-learning applications, large datasets are divided into two parts: 80% training dataset and 20% test dataset. A classifier or a regressor will be fitted using the 80% training dataset and the performance of the model will be evaluated on the remaining test dataset. For small datasets, *k*-fold cross validation techniques are generally used, where the whole dataset is resampled *k* times to train the model *k* times and evaluate its performance on the unused fold. Although the cross validation techniques allow a proper estimation of the generalization performance of the constructed model, its use in deep learning is limited due to the large training time and memory requirement. Thus, in deep learning applications the dataset is mostly divided into three parts: training, test and validation dataset. The training dataset is used to fit the deep learning model. The validation dataset provides an unbiased evaluation of the fitted deep learning model and simultaneously optimizes the hyperparameters of the model. And finally, the test dataset is used for evaluating the performance of the final model fitted on the training dataset. The division of the dataset into parts should be made at the highest hierarchical level. For instance, in a clinical setting, the highest hierarchical level is at the patient-level or device-level. Images or spectra obtained from the same patient should be a part of either the training, validation or the test dataset, to avoid any training bias [145]. A training bias is introduced when both the training and validation dataset originate from the same source (patient or device), thus reaching a high training and validation accuracy but a poor test accuracy. In prospect, splitting the dataset plays a major role in training deep learning models. Thus, it is beneficial for the biophotonic community to encourage proper model validation.

5.2.4 | Model validation and assessment of model performance

Establishing common procedures for model validation is important for biophotonics as explained in section 5.1.5. This facilitates a fair comparison between different models and systems. It is a common practice to test a final model on a third "independent test set" (also referred to as "holdout set") beside the "training set" and the "validation set." The latter mainly serves the purpose of model selection and hyperparameter optimization [4, 7, 8]. However, this requires a lot of data which represents the whole underlying population. To deal with small datasets cross-validation using the *k*-fold strategy is a commonly used approach. [145]

While training a deep neural network, the accuracy on training and validation dataset rises gradually with the number of iterations. If not, then several possibilities are responsible to lower the performance including over-fitting of the model on the training dataset, a small dataset size, a noisy dataset, the choice of hyperparameters and the depth of the model. In such cases, increasing the dataset by data-augmentation techniques, removing redundant data by filtering noisy images or spectra, optimizing the hyperparameters and performing cross validation can be considered. Nevertheless, reducing over-fitting requires systematic studies depending on the dataset.

In addition to the above-mentioned techniques, early-stopping of the model training can also be utilized to improve the generalization performance [146, 147]. Early stopping is a regularization technique that stops the training of the deep learning models before the performance on the validation dataset begins to decline. In cross validation of deep learning the model with the best validation accuracy can be used to predict the test data. In the case of comparison of two or more models, the performances on test dataset should be reported.

5.2.5 | Reduce over-fitting

As explained earlier (see section 5.1.3), a deep learning model trained with high variance can predict well on the training data but shows a poor generalizability to the test data. Adjusting the generalizability and constructing robust models is done by reducing over-fitting. This is often termed as "regularization" [6, 142] and can be achieved by several methods. Augmentation of training data explained in section 5.2.1 is often considered as one of the regularization methods [148]. Another method is to add dropout layers to the model. Adding dropout layer is based on the principle: "learn less to learn better." In this regularization technique, the outputs of some neurons in the hidden layers are ignored, thereby, forcing the remaining neurons to learn a sparse representation of the data [149, 150]. Several variations of the dropout method reported in the literature have shown to improve model performances [151–154]. In addition to the dropout methods, early stopping (explained in section 5.2.4) and weight regularization techniques are other regularization methods for reducing over-fitting.

Weight regularization like L1 and L2 regularization penalizes the model during training based on the magnitude of the learned weights [155, 156], because large weights of a deep neural network can be a sign of an unstable network [157]. Regularization techniques encourage the sum of absolute values of the weights (L1) or sum of squared values of the weights (L2) to be minimum and thereby generating sparse weights that reduce over-fitting. Another method to check over-fitting is to reduce the capacity of deep learning models by decreasing the number of layers in the model or number of parameters in each layer [4].

Besides these regularization techniques, batch normalization technique is a well-known method to overcome over-fitting of deep neural networks [158]. This technique standardizes the inputs to a layer of deep neural network for each mini-batch. In this way, training of the deep neural network is stabilized and the training process is accelerated [159].

In summary, all the earlier explained topics are complementary to each other with a common goal of reducing over-fitting and constructing robust deep learning models. However, the effects of each of these regularization methods on biophotonic data need systematic investigation.

6 | CONCLUSION AND FUTURE OUTLOOK

Biophotonics is a rapidly growing field with a great potential to be a part of clinical practice. Current technological advancements in biophotonics are pushing the limits by increasing the resolution of optical systems, achieving larger penetration depths and faster scanning speeds. Additionally, current optical systems are capable of probing from micro to macroscopic scales, detectors are becoming more specific and efforts for miniaturizing devices using fibers are observed [3, 160]. All these technological advancements are enriching the information content of the biophotonic data and advanced data analysis methods, like deep learning techniques, are needed. In this regard, researchers are developing deep learning methods for various biophotonic applications, which were elaborated in this review article.

Out of all the contributions discussed in this review article, a majority of work includes deep learning methods for biophotonic image data, whereas deep learning for spectral data is still underexplored. Almost 60% of the research used image data for early detection of diseases and assessment of disease stages. The remaining work majorly focused on virtual staining, increasing the resolution of fluorescence images and segmentation of cells, tissues and organs in spectroscopic images. In addition, a small part of the reviewed papers focused on preprocessing and classification of vibrational spectroscopic data. Although deep learning methods are underexplored for spectral data, we foresee that its development for vibrational spectroscopic data can transform the biophotonics field. Therefore, we discuss some potential applications of deep learning to analyze image and spectral data in this review.

Deep learning architectures can be used for spectral classification without the need of complex preprocessing steps [100]. On the other side, architectures like RNNs can be used for spectral preprocessing including denoising or despiking. Due to the basic similarities in the shape of the spectra, classification models can be trained with spectral data obtained from different domains using transfer-learning methods [100]. We speculate that transfer learning can complement the model-transfer methods [161] built for spectroscopic data by transferring high-level features of training data obtained in one domain to new data acquired in another domain. Until now, transfer learning methods have proven beneficial for fluorescence imaging data especially for cases where large datasets were not available [26, 74, 75].

Deep learning for vibrational spectroscopy has some challenges like the lack of data, the complexity of spectra, inter and intra-class-variances within the spectra and interpretability of the deep learning models. The issue of lack of data can be addressed by creating and facilitating access to large databases of spectroscopic data and efforts have already been initialized in this direction. Recent studies have reported large databases comprising images of three modalities including confocal, two-photon and wide-field fluorescence microscopy depicting biological samples [162–164]. Along with creating large databases, it is equally important to adopt standardized data acquisition protocols, acquire balanced datasets and reliable annotations to increase the current state-of-the-art performances of the models. In order to achieve robust and reliable deep learning models and to use them in clinical setting, it is required to apply online training, updating the model parameters with the arrival of new data and check the data and model reproducibility. At the same time, the biophotonic community should adopt validating standards in order to avoid publishing over-fitted deep learning models. Despite of the outstanding progress of deep learning methods in biophotonics field, their reliability as decision-making systems is always questionable due to their "black-box" behavior. Thus, researchers are developing methods to understand the deep learning predictions [136, 165]. Nevertheless, this topic needs more investigations.

Finally, we have to answer our initial question "Is deep learning a boon for biophotonics?" We think that deep learning is eventually going to be a boon to biophotonics, which will revolutionize the decision-making approaches for pathologist, clinicians and doctors. A motivating example of deep learning used in optical systems is the IDx-DR device, a clinically accepted deep learning model to detect diabetes retinopathy in optical coherence tomography images [166]. Another potential example is GAN-based modeling for virtual staining of autofluorescence images which can bypass the long staining protocols and help the pathologist to compare new biophotonic technologies with the "gold-standard" staining methods. However, deep learning for biophotonics is still in an infant stage and requires overcoming various hurdles before coming into clinical usage. A large amount of data, quality check for the data, providing reliable annotations, appropriate model validation, interpreting model predictions and improving hardware capacities are vital for overcoming these hurdles. Overcoming these challenges and achieving optimal decision-making algorithms based on deep learning for modern healthcare systems is potentially the future of biophotonics.

ACKNOWLEDGMENTS

Funding of the German Research Foundation (DFG) for the projects BO 4700/1-1, BO 4700/4-1, PO 563/30-1 and STA 295/11-1 is highly acknowledged. This work received financial support by the Ministry for Economics, Sciences and Digital Society of Thuringia (TMWWDG), under the framework of the Landesprogramm ProDigital (DigLeben-5575/10-9).

CONFLICT OF INTEREST

The authors declare no conflicts of interest.

AUTHOR CONTRIBUTIONS

The manuscript was written through contributions of all authors. All authors have given approval to the final version of the manuscript.

ORCID

Pranita Pradhan https://orcid.org/0000-0002-0558-2914

Thomas W. Bocklitz https://orcid.org/0000-0003-2778-6624

REFERENCES

[1] C. Krafft, *J. Biophotonics* **2016**, *9*(11-12), 1362.
[2] N. Vogler, S. Heuke, T. W. Bocklitz, M. Schmitt, J. Popp, *Annu. Rev. Anal. Chem.* **2015**, *8*, 359.
[3] L. Marcu, S. A. Boppart, M. R. Hutchinson, J. Popp, B. C. Wilson, *J. Biomed. Opt.* **2017**, *23*(2), 021103.
[4] C. M. Bishop, *Neural Networks for Pattern Recognition*, Oxford University Press, Oxford **1995**.
[5] M. P. McBee, O. A. Awan, A. T. Colucci, C. W. Ghobadi, N. Kadom, A. P. Kansagra, S. Tridandapani, W. F. Auffermann, *Acad. Radiol.* **2018**, *25*(11), 1472.
[6] I. Goodfellow, Y. Bengio, A. Courville, *Deep Learning*, MIT Press, Cambridge, MA **2016**.
[7] B. D. Ripley, *Pattern Recognition and Neural Networks*, Cambridge University Press, Cambridge **1996**.
[8] Stuart Russel, Peter Norvig, EUA: Prentice Hall **2003**, *178*.
[9] F. Rosenblatt, *Psychol. Rev.* **1958**, *65*(6), 386.
[10] B. R. Upadhyaya, G. Mathai, E. Eryurek, in *29th IEEE Conference on Decision and Control*, IEEE, **1990**, pp. 3277-3282.
[11] T. Bocklitz, M. Putsche, C. Stüber, J. Käs, A. Niendorf, P. Rösch, J. Popp, *J. Raman Spectrosc.* **2009**, *40*(12), 1759.
[12] W. S. McCulloch, W. Pitts, *Bull. Math. Biophys.* **1943**, *5*(4), 115.
[13] D. H. Hubel, T. N. Wiesel, *J. Physiol.* **1963**, *165*(3), 559.
[14] S. Brahnam, L. C. Jain, L. Nanni, A. Lumini, *Local Binary Patterns: New Variants and Applications*, Springer, Berlin, Heidelberg **2013**.
[15] Léon Bottou, *On-Line Learning in Neural Networks*, Cambridge University Press, New York **1998**, 17 (9), 142.
[16] L. Bottou, in *Proceedings of COMPSTAT'2010*, Springer, **2010**, pp. 177-186.
[17] D. E. Rumelhart, G. E. Hinton, R. J. Williams, *Nature* **1986**, *323*(6088), 533.
[18] Y. LeCun, B. E. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. E. Hubbard, L. D. Jackel, in *Advances in Neural Information Processing Systems*, **1990**, pp. 396-404.
[19] D. P. Kingma, J. Ba, *arXiv preprint arXiv:1412.6980* **2014**.
[20] M. D. Zeiler, *arXiv preprint arXiv:1212.5701* **2012**.
[21] J. Duchi, E. Hazan, Y. Singer, *J. Mach. Learn. Res.* **2011**, *12*, 2121.
[22] Y. Le Cun, L. D. Jackel, B. Boser, J. S. Denker, H. P. Graf, I. Guyon, D. Henderson, R. E. Howard, W. Hubbard, *IEEE Commun. Mag.* **1989**, *27*(11), 41.
[23] A. F. Agarap, *arXiv preprint arXiv:1803.08375* **2018**.
[24] B. Karlik, A. V. Olgac, *Int. J. Artif. Intell. Expert Syst.* **2011**, *1* (4), 111.
[25] C. Nwankpa, W. Ijomah, A. Gachagan, S. Marshall, *arXiv preprint arXiv:1811.03378* **2018**.
[26] H. Lin, C. Wei, G. Wang, C. Hu, L. Lin, M. Ni, J. Chen, S. Zhuo, *J. Biophotonics* **2019**, *12*(7), e201800435.
[27] H. Fan, F. Zhang, X. Liang, Z. Li, G. Liu, Y. Xu, *J. Biophotonics* **2019**, *12*(7), e201800488.
[28] Z. C. Lipton, J. Berkowitz, C. Elkan, *arXiv preprint arXiv:1506.00019* **2015**.
[29] J. Mao, W. Xu, Y. Yang, J. Wang, Z. Huang, A. Yuille, *arXiv preprint arXiv:1412.6632* **2014**.

[30] R. Socher, A. Perelygin, J. Y. Wu, J. Chuang, C. D. Manning, A. Y. Ng, C. Potts, *EMNLP* **1631**, *2013*, 1631.

[31] K. Cho, B. van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, Y. Bengio, in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Association for Computational Linguistics, Doha, Qatar, **2014**, pp. 1724-1734. https://www.aclweb.org/anthology/D14-1179.

[32] W. Zaremba, I. Sutskever, O. Vinyals, *arXiv preprint arXiv: 1409.2329* **2014**.

[33] P. Baldi, G. Pollastri, *J. Mach. Learn. Res.* **2003**, *4*, 575.

[34] X. Zhu, P. Sobihani, H. Guo, in *International Conference on Machine Learning*, **2015**, pp. 1604-1612.

[35] D. H. Ballard, in *Proceedings of the Sixth National Conference on Artificial Intelligence – Volume 1*, AAAI Press, of *AAAI'87*, **1987**, pp. 279-284.

[36] H. Bourlard, Y. Kamp, *Biol. Cybern.* **1988**, *59*, 291.

[37] P. Pradhan, T. Meyer, M. Vieth, A. Stallmach, M. Waldner, M. Schmitt, J. Popp, T. Bocklitz, in *Proceedings of the 8th International Conference on Pattern Recognition Applications and Methods – Volume 1: ICPRAM*, INSTICC, SciTePress, **2019**, pp. 396-405.

[38] L. Duan, X. Qin, Y. He, X. Sang, J. Pan, T. Xu, J. Men, R. E. Tanzi, A. Li, Y. Ma, C. Zhou, *CoRR* **2018**, *abs/1803.01947*.

[39] G. Litjens, T. Kooi, B. E. Bejnordi, A. A. A. Setio, F. Ciompi, M. Ghafoorian, J. A. Van Der Laak, B. Van Ginneken, C. I. Sánchez, *Med. Image Anal.* **2017**, *42*, 60.

[40] K. Raza, N. K. Singh, *arXiv preprint arXiv:1812.07715* **2018**.

[41] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, Y. Bengio, Generative adversarial nets. in *Advances in Neural Information Processing Systems 27* (Eds: Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, K. Q. Weinberger), Curran Associates, Inc., New York **2014**, p. 2672.

[42] Y. Rivenson, Z. Göröcs, H. Günaydin, Y. Zhang, H. Wang, A. Ozcan, *Optica* **2017**, *4*(11), 1437.

[43] H. Wang, Y. Rivenson, Y. Jin, Z. Wei, R. Gao, H. Günaydın, L. A. Bentolila, C. Kural, A. Ozcan, *Nat. Methods* **2019**, *16* (1), 103.

[44] Y. Rivenson, H. Wang, Z. Wei, K. Haan, Y. Zhang, Y. Wu, H. Gunaydin, J. Zuckerman, T. Chong, A. Sisk, L. Westbrook, W. Wallace, A. Ozcan, *Nat. Biomed. Eng.* **2019**, *3*, 466.

[45] C. Fu, S. Lee, D. J. Ho, S. Han, P. Salama, K. W. Dunn, E. J. Delp, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, **2018**, pp. 2221-2229.

[46] A. Osokin, A. Chessel, R. E. Carazo Salas, F. Vaggi, in *Proceedings of the IEEE International Conference on Computer Vision*, **2017**, pp. 2233-2242.

[47] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, T. Darrell, in *Proceedings of the 22nd ACM International Conference on Multimedia*, **2014**, pp. 675-678.

[48] R. Collobert, S. Bengio, J. Mariéthoz, *Torch: A Modular Machine Learning Software Library*, Idiap, Martigny, Switzerland **2002**.

[49] The Team, R. Al-Rfou, G. Alain, A. Almahairi, C. Angermueller, D. Bahdanau, N. Ballas, F. Bastien, J. Bayer, A. Belikov, A. Belopolsky, Y. Bengio, A. Bergeron, J. Bergstra, V. Bisson, J. B. Snyder, N. Bouchard, N. Boulanger-Lewandowski, X. Bouthillier, Y. Zhang, *arXiv preprint arXiv:1605.02688* **2016**.

[50] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems. Retrieved from https://www.tensorflow.org/, **2015**.

[51] F. Chollet, and others *Keras*, https://keras.io, **2015**.

[52] S. Dieleman, J. Schlüter, C. Raffel, E. Olson, S. K. Sønderby, D. Nouri, D. Maturana, M. Thoma, E. Battenberg, J. Kelly, J. De Fauw, M. Heilman, D. M. de Almeida, B. McFee, H. Weideman, G. Takács, P. de Rivaz, J. Crall, G. Sanders, K. Rasul, C. Liu, G. French, J. Degrave, *Lasagne: First Release*, **2015**. https://doi.org/10.5281/zenodo.27878.

[53] M. Nielsen, Y. Bengio, A. Couville, Retrieved from http://neuralnetworksanddeeplearning **2017**.

[54] C. M. Bishop, *Pattern Recognition and Machine Learning*, Springer, New York **2006**.

[55] L. Deng, D. Yu, *Foundations and Trends in Signal Process.* **2014**, *7*(3-4), 197.

[56] K. Gurney, *An Introduction to Neural Networks*, CRC Press, London, UK **2014**.

[57] H. D. Beale, H. B. Demuth, M. T. Hagan, *PWS Publish Company* **1996**, *11*, 1–47.

[58] S. Haykin, *Neural Networks and Learning Machines, 3/E*, Pearson Education India, Hoboken, NJ **2010**.

[59] K. J. Halupka, B. J. Antony, M. H. Lee, K. A. Lucy, R. S. Rai, H. Ishikawa, G. Wollstein, J. S. Schuman, R. Garnavi, *Biomed. Opt. Express* **2018**, *9*(12), 6205.

[60] Y. Ma, X. Chen, W. Zhu, X. Cheng, D. Xiang, F. Shi, *Biomed. Opt. Express* **2018**, *9*(11), 5129.

[61] Y. Huang, Z. Lu, Z. Shao, M. Ran, J. Zhou, L. Fang, Y. Zhang, *Opt. Express* **2019**, *27*(9), 12289.

[62] A. Janowczyk, A. Madabhushi, *J. Pathol. Inf.* **2016**, *7*(1), 29.

[63] X. Xu, J. Cheng, M. Thrall, Z. Liu, X. Wang, S. Wong, *Biomed. Opt. Express* **2013**, *4*, 2855.

[64] Y. Li, R. Zheng, Y. Wu, K. Chu, Q. Xu, M. Sun, Z. J. Smith, *J. Biophotonics* **2019**, *12*(9), e201800410.

[65] O. Ronneberger, P. Fischer, T. Brox, in *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, Springer, Vol. 9351 of *LNCS*, **2015**, pp. 234-241 (available on arXiv:1505.04597 [cs.CV]). http://lmb.informatik.uni-freiburg.de/Publications/2015/RFB15a.

[66] V. Badrinarayanan, A. Kendall, R. Cipolla, *IEEE Trans. Pattern Anal. Mach. Intell.* **2015**, *39*, 2481.

[67] F. Visin, K. Kastner, A. C. Courville, Y. Bengio, M. Matteucci, K. H. Cho, *CoRR* **2015**, *abs/1511.07053*.

[68] J. Gabriel, J. S. Brostow, J. Fauqueur, R. Cipolla, in *European Conference on Computer Vision*, Springer, **2008**, pp. 44-57.

[69] Y. Xie, Z. Zhang, M. Sapkota, Y. Lin, *Medical Image Computing and Computer-assisted Intervention: MICCAI ... International Conference on Medical Image Computing and Computer-Assisted Intervention* **2016**, *9901*, 185-193.

[70] G. Jun, F. C. Yaw, B. K. Ng, S. G. Razul, S. K. Lim, *J. Biophotonics* **2014**, *7*(7), 483.

[71] E. Ryzhikova, O. Kazakov, L. Halamkova, D. Celmins, P. Malone, E. Molho, E. A. Zimmerman, I. K. Lednev, *J. Biophotonics* **2015**, *8*(7), 584.

[72] G. I. Gopakumar, M. Swetha, S. S. Gorthi, G. S. Subrahmanyam, *J. Biophotonics* **2017**, *11*, e201700003.

[73] K. Aljakouch, Z. Hilal, I. Daho, M. Schuler, S. D. Krauß, H. K. Yosef, J. Dierks, A. Mosig, K. Gerwert, S. F. El-Mashtoly, *Anal. Chem.* **2019**, *91*(21), 13900.

[74] S. Weng, X. Xu, J. Li, S. T. C. Wong, *J. Biomed. Opt.* **2017**, *22* (10), 106017.

[75] N. Singla, K. Dubey, V. Srivastava, *J. Biophotonics* **2019**, *12* (3), e201800255.

[76] N. Ali, E. Quansah, K. Köhler, T. Meyer, M. Schmitt, J. Popp, A. Niendorf, T. Bocklitz, *Transl. Biophotonics* **2019**, *1*(1-2), e201900003.

[77] C. Szegedy, Wei Liu, Yangqing Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, A. Rabinovich, in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, **2015**, pp. 1-9.

[78] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, Z. Wojna, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, **2016**, pp. 2818-2826.

[79] K. Simonyan, A. Zisserman, *arXiv 1409.1556* **2014**.

[80] A. E. Heidari, T. T. Pham, I. Ifegwu, R. Burwell, W. B. Armstrong, T. Tjoson, S. Whyte, C. Giorgioni, B. Wang, B. J. F. Wong, Chen, Z, *J. Biophotonics* **2019**, *13* (3), e201900221.

[81] J. Deng, W. Dong, R. Socher, L. Li, K. Li, L. Fei-Fei, in *2009 IEEE Conference on Computer Vision and Pattern Recognition*, 2009, pp. 248-255.

[82] K. He, X. Zhang, S. Ren, J. Sun, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, **2016**, pp. 770-778.

[83] X. Feng, K. Qing, N. J. Tustison, C. H. Meyer, Q. Chen, *Med. Phys.* **2019**, *46*(5), 2169.

[84] Y. Liu, W. Fu, V. Selvakumaran, M. Phelan, W. Paul Segars, E. Samei, M. Mazurowski, J. Y. Lo, G. D. Rubin, R. Henao, in *Medical Imaging 2019: Imaging Informatics for Healthcare, Research, and Applications* (Eds: P.-H. Chen, P. R. Bak), International Society for Optics and Photonics, SPIE **2019**, p. 319. https://doi.org/10.1117/12.2512887.

[85] N. Bayramoglu, M. Kaakinen, L. Eklund, J. Heikkila, in *Proceedings of the IEEE International Conference on Computer Vision Workshops*, **2017**, pp. 64-71.

[86] Z. Xu, C. F. Moro, B. Bozóky, Q. Zhang, *arXiv preprint arXiv: 1901.04059* **2019**.

[87] T. Gregory Schaaff, J. M. Mcmahon, P. J. Todd, *Anal. Chem.* **2002**, *74*(17), 4361.

[88] T. Bocklitz, A. Crecelius-Vitz, C. Matthäus, N. Tarcea, F. von Eggeling, M. Schmitt, U. Schubert, J. Popp, *Anal. Chem.* **2013**, *85*(22), 10829.

[89] J. T. Kwak, S. Hewitt, S. Sinha, R. Bhargava, *BMC Cancer* **2011**, *11*, 62.

[90] C. Yang, D. Niedieker, F. Grosserueschkamp, M. Horn, A. Tannapfel, A. Kallenbach-Thieltges, K. Gerwert, A. Mosig, *BMC Bioinformatics* **2015**, *16*(1), 396.

[91] H. Uzunova, M. Wilms, H. Handels, J. Ehrhardt, in *International Conference on Medical Image Computing and Computer-Assisted Intervention*, **2017**, pp. 223-231.

[92] H. Li, Y. Fan, *arXiv preprint arXiv:1709.00799* **2017**.

[93] E. Nehme, L. E. Weiss, T. Michaeli, Y. Shechtman, *Optica* **2018**, *5*(4), 458.

[94] Z. Luo, A. Yurt, R. Stahl, A. Lambrechts, V. Reumers, D. Braeken, L. Lagae, *Opt. Express* **2019**, *27*(10), 13581.

[95] O. Ryabchykov, S. Guo, T. Bocklitz, *Phys. Sci. Rev.* **2018**, *4*(2), 20170043.

[96] S. Seifert, V. Merk, J. Kneipp, *J. Biophotonics* **2016**, *9* (1-2), 181.

[97] V. Tafintseva, E. Vigneau, V. Shapaval, V. Cariou, E. M. Qannari, A. Kohler, *J. Biophotonics* **2018**, *11*(3), e201700047.

[98] H. Chen, X. Li, N. Broderick, Y. Liu, Y. Zhou, J. Han, W. Xu, *J. Biophotonics* **2018**, *11*(9), e201800016.

[99] J. Gerretzen, E. Szymańska, J. Jansen, J. Bart, H.-J. Manen, E. Van den Heuvel, L. Buydens, *Anal. Chem.* **2015**, *87*(24), 12096.

[100] J. Liu, M. Osadchy, L. Ashton, M. Foster, C. J. Solomon, S. J. Gibson, *Analyst* **2017**, *142*(21), 4067.

[101] E. J. Bjerrum, M. Glahder, T. Skov, *arXiv preprint arXiv: 1710.01927* **2017**.

[102] A. Raulf, J. Butke, C. Küpper, F. Großerüschkamp, K. Gerwert, A. Mosig, *Bioinformatics (Oxford, England)* **2020**, *36*(1), 287.

[103] J. Acquarelli, T. van Laarhoven, J. Gerretzen, T. Tran, L. Buydens, E. Marchiori, *Anal. Chim. Acta* **2016**, *954*, 22.

[104] J. Dong, M. Hong, Y. Xu, X. Zheng, *J. Chemometr.* **2019**, *33* (11), e3184.

[105] H. Lin, F. Deng, K.-C. Huang, H. J. Lee, J.-X. Cheng, *CLEO: Applications and Technology*, Optical Society of America, San Jose, CA **2019**, ATu3K.

[106] H. He, M. Xu, Z. Cheng, P. Zheng, L. Luo, L. Wang, B. Ren, *Anal. Chem.* **2019**, *91*, 7070.

[107] Y. Bengio, A. Courville, P. Vincent, *IEEE Trans. Pattern Anal. Mach. Intell.* **2013**, *35*(8), 1798.

[108] T. Trakoolwilaiwan, B. Behboodi, J. Lee, K. Kim, J.-W. Choi, *Neurophotonics* **2017**, *5*, 1.

[109] S. Hiwa, K. Hanawa, R. Tamura, K. Hachisuka, T. Hiroyasu, *Comput. Intell. Neurosci.* **2016**, *2016*, 1.

[110] M. Yu, H. Yan, J. Xia, L. Zhu, T. Zhang, Z. Zhu, X. Lou, G. Sun, M. Dong, *Photodiagn. Photodyn. Ther.* **2019**, *26*, 430.

[111] W. B. Sohn, S. Y. Lee, S. Kim, *J. Raman Spectrosc.* **2020**, *51* (3), 414.

[112] S. Krauß, R. Roy, H. Yosef, T. Lechtonen, S. El-Mashtoly, K. Gerwert, A. Mosig, *J. Biophotonics* **2018**, *11*(10), e201800022.

[113] C.-S. Ho, N. Jean, C. A. Hogan, L. Blackmon, S. S. Jeffrey, M. Holodniy, N. Banaei, A. A. E. Saleh, S. Ermon, J. Dionne, *Nat. Commun.* **2019**, *10*(1), 1.

[114] Salim Malek, Farid Melgani, Yakoub Bazi, J. Chemometr. **2017**.

[115] X. Yu, L. Tang, X. Wu, H. Lu, *Food Anal. Methods* **2017**, *11*, 1.

[116] Z. Jianqiang, L. Weijuan, H. Ying, Q. Changgui, Y. Shuangyan, L. Changyu, N. Linru, *Anal. Lett.* **2018**, *51*(7), 1029.

[117] B. Le, X. Dong, Y. Mao, D. He, *Infrared Phys. Technol.* **2018**, *93*, 34.

[118] X. Fan, W. Ming, H. Zeng, Z. Zhang, H. Lu, *Analyst* **2019**, *144*, 1789.

[119] K. Liland, A. Kohler, N. Afseth, *J. Raman Spectrosc.* **2016**, *47*, 643.

[120] D. Soekhoe, P. Van Der Putten, A. Plaat, in *International Symposium on Intelligent Data Analysis*, Springer, **2016**, pp. 50-60.

[121] J. Cho, K. Lee, E. Shin, G. Choy, S. Do, *arXiv preprint arXiv:1511.06348* **2015**.

[122] C. Shorten, T. M. Khoshgoftaar, *J. Big Data* **2019**, *6*(1), 60.

[123] L. Engstrom, B. Tran, D. Tsipras, L. Schmidt, A. Madry, *arXiv preprint arXiv:1712.02779* **2017**.

[124] J. Yosinski, J. Clune, Y. Bengio, H. Lipson, in *Advances in Neural Information Processing Systems*, **2014**, pp. 3320-3328.

[125] M. Buda, A. Maki, M. A. Mazurowski, *Neural Netw.* **2018**, *106*, 249.

[126] J. M. Johnson, T. M. Khoshgoftaar, *J. Big Data* **2019**, *6* (1), 27.

[127] S. Wang, W. Liu, J. Wu, L. Cao, Q. Meng, P. J. Kennedy, in *2016 international Joint Conference on Neural Networks (IJCNN)*, IEEE, *2016*, pp. 4368-4374.

[128] B. Neal, S. Mittal, A. Baratin, V. Tantia, M. Scicluna, S. Lacoste-Julien, I. Mitliagkas, *arXiv preprint arXiv:1810.08591* **2018**.

[129] T. Ching, D. S. Himmelstein, B. K. Beaulieu-Jones, A. A. Kalinin, B. T. Do, G. P. Way, E. Ferrero, P.-M. Agapow, M. Zietz, M. M. Hoffman, W. Xie, G. L. Rosen, B. J. Lengerich, J. Israeli, J. Lanchantin, S. Woloszynek, A. E. Carpenter, A. Shrikumar, J. Xu, E. M. Cofer, C. A. Lavender, S. C. Turaga, A. M. Alexandari, Z. Lu, D. J. Harris, D. DeCaprio, Y. Qi, A. Kundaje, Y. Peng, L. K. Wiley, M. H. S. Segler, S. M. Boca, S. Joshua Swamidass, A. Huang, A. Gitter, C. S. Greene, *J. R. Soc. Interface* **2018**, *15*(141), 20170387.

[130] C. Zhang, S. Bengio, M. Hardt, B. Recht, O. Vinyals, *arXiv preprint arXiv:1611.03530* **2016**.

[131] S. Bach, A. Binder, G. Montavon, F. Klauschen, K.-R. Müller, W. Samek, *PLoS One* **2015**, *10*(7), e0130140.

[132] G. Montavon, W. Samek, K.-R. Müller, *Dig. Signal Process.* **2018**, *73*, 1.

[133] G. Montavon, S. Lapuschkin, A. Binder, W. Samek, K.-R. Müller, *Pattern Recogn.* **2017**, *65*, 211.

[134] W. Samek, T. Wiegand, K.-R. Müller, *CoRR* **2017**, *abs/1708.08296*.

[135] K. Simonyan, A. Vedaldi, A. Zisserman, *preprint* **2013**.

[136] T. Bocklitz, in *Proceedings of the 8th International Conference on Pattern Recognition Applications and Methods – Volume 1* ICPRAM, INSTICC, SciTePress, 2019, pp. 874-880.

[137] R. Horstmeyer, R. Heintzmann, G. Popescu, L. Waller, C. Yang, *Nat. Photonics* **2016**, *10*(2), 68.

[138] M. Koch, P. Symvoulidis, V. Ntziachristos, *Nat. Photonics* **2018**, *12* (9), 505.

[139] *Nat. Photonics* **2018**, *12*, 117.

[140] Z. Hussain, F. Gimenez, D. Yi, D. Rubin, *AMIA Annual Symposium Proceedings* **2017**, 979-984.

[141] A. Mikołajczyk, M. Grochowski, *Int. Interdisc. PhD Workshop (IIPhDW)* **2018**, *2018*, 117.

[142] C. M. Bishop, *Neural Comput.* **1995**, *7*, 108.

[143] L. Perez, J. Wang, *CoRR* **2017**, *abs/1712.04621*.

[144] S. J. Pan, Q. Yang, *IEEE Trans. Knowl. Data Eng.* **2009**, *22* (10), 1345.

[145] S. Guo, T. Bocklitz, U. Neugebauer, J. Popp, *Anal. Methods* **2017**, *9*, 4410.

[146] L. Prechelt, *Neural Netw.* **1998**, *11*(4), 761.

[147] L. Prechelt, in *Early Stopping—But When?* (Eds: G. Montavon, G. B. Orr, K.-R. Müller), Springer, Berlin Heidelberg, Berlin, Heidelberg **2012**, p. 53.

[148] J. Kukačka, V. Golkov, D. Cremers, *arXiv preprint arXiv:1710.10686* **2017**.

[149] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, R. Salakhutdinov, *J. Mach. Learn. Res.* **2014**, *15*(1), 1929.

[150] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, R. R. Salakhutdinov, *arXiv preprint arXiv:1207.0580* **2012**.

[151] J. Ba, B. Frey, in *Advances in Neural Information Processing Systems*, **2013**, pp. 3084-3092.

[152] X. Bouthillier, K. Konda, P. Vincent, R. Memisevic, *arXiv preprint arXiv:1506.08700* **2015**.

[153] Shin-ichi Maeda, *arXiv preprint arXiv:1412.7003* **2014**.

[154] Y. Gal, Z. Ghahramani, in *International Conference on Machine Learning*, **2016**, pp. 1050-1059.

[155] D. C. Plaut, S. J. Nowlan, G. E. Hinton, *Experiments on Learning by Back Propagation*. ERIC, Washington, DC **1986**.

[156] A. Krogh, J. A. Hertz, in *Advances in Neural Information Processing Systems 4* (Eds: J. E. Moody, S. J. Hanson, R. P. Lippmann), Morgan-Kaufmann, Santa Cruz, CA **1992**, p. 950.

[157] R. Reed, R. J. MarksII, *Neural Smithing: Supervised Learning in Feedforward Artificial Neural Networks*, MIT Press, Cambridge, MA **1999**.

[158] Y. LeCun, L. Bottou, G. B. Orr, K. R. Müller, in *Efficient BackProp* (Eds: G. B. Orr, K.-R. Müller), Springer, Berlin Heidelberg, Berlin, Heidelberg **1998**, p. 9.

[159] S. Ioffe, C. Szegedy, *arXiv preprint arXiv:1502.03167* **2015**.

[160] C. Brian, M. J. Wilson, F. Leblond, *J. Biomed. Opt.* **2018**, *23* (3), 030901.

[161] S. Guo, R. Heinke, S. Stöckel, P. Rösch, J. Popp, T. Bocklitz, *J. Raman Spectrosc.* **2018**, *49*(4), 627.

[162] M. Riffle, T. N. Davis, *BMC Bioinformatics* **2010**, *11*(1), 263.

[163] Y. Zhang, Y. Zhu, E. Nichols, Q. Wang, S. Zhang, C. Smith, S. Howard, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, **2019**, pp. 11710.

[164] E. Czech, B. A. Aksoy, P. Aksoy, J. Hammerbacher, *BMC Bioinformatics* **2019**, *20*(1), 1.

[165] J. de La Torre, A. Valls, D. Puig, *Neurocomputing* **2019**.

[166] M. D. Abràmoff, Y. Lou, A. Erginay, W. Clarida, R. Amelon, J. C. Folk, M. Niemeijer, *Investig. Ophthalmol. Vis. Sci.* **2016**, *57*(13), 5200.

How to cite this article: Pradhan P, Guo S, Ryabchykov O, Popp J, Bocklitz TW. Deep learning a boon for biophotonics? *J. Biophotonics*. 2020;13:e201960186. https://doi.org/10.1002/jbio.201960186