OPTICS CONTINUUM

On the relevance of hyperparameter optimization of convolutional neural networks for retrieving spectral signals

ALI SAGHI,* LASSE LENSU, AND ERIK M. VARTIAINEN

LUT School of Engineering Science, LUT University, 53851 Lappeenranta, Finland
*ali.saghi.2015@gmail.com

Abstract: Coherent anti-Stokes Raman scattering (CARS) is a third-order nonlinear optical process used in spectroscopy to analyze molecular structures. One significant drawback of this approach is its non-resonant background contribution, which distorts spectral line shapes and consequently diminishes the accuracy of chemical information. A state-of-the-art solution for automatically extracting the Raman signals from CARS spectra is the convolutional neural network (CNN) model. In this research, we studied the use of hyperparameter optimization of SpecNet, a CNN model proposed in the literature, to improve the extraction of the Raman signal from CARS spectra. The original SpecNet has two major problems: first, the model is incapable of recovering spectral peaks near the edges of the spectral range, and second, it cannot match the intensity of the peaks throughout the spectrum. In this work, these two problems were successfully mitigated by optimization of the hyperparameters of the SpecNet model.

1. Introduction

Coherent anti-Stokes Raman scattering (CARS) is a third-order nonlinear optical process that is resonantly enhanced by molecular vibrations, providing label-free and nondestructive information about the investigated molecule at high speeds (approx. 5×10^4 spectra per second) [1–3]. It is a four-wave mixing process where two laser fields ("pump" at frequency ω_{pu} and "Stokes" at ω_{S}) coherently drive a Raman transition, which is probed by a third field ("probe" at ω_{pr}), thus generating the anti-Stokes signal at $\omega_{\mathrm{as}} = (\omega_{\mathrm{pu}} - \omega_{\mathrm{S}}) + \omega_{\mathrm{pr}}$. If the frequency difference (ω_{pu} - ω_{S}) matches a transition in the sample, the CARS signal is greatly enhanced.

In CARS imaging, the imaging structures are created by visualizing the vibrational contrast of the molecules that, on the one hand, needs a fraction of a second to generate the micrograph of a specific vibrational mode and, on the other hand, requires a few minutes to produce the hyperspectral image. CARS imaging has found applications in various areas, including breast cancer tissue imaging [4], *in vivo* imaging of biological cells [5], and understanding of lipid biology [6].

A CARS spectrum is a combination of a resonant signal and a non-resonant background (NRB) and although this NRB is a pure real and constant value, since these two contributions are coherently combined, it is not possible to remove the NRB by just subtraction. The NRB strengthens the CARS intensity compared to the spontaneous Raman scattering in magnitude as a stable homodyne amplifier but simultaneously perturbs the measured CARS signal, thus distorting the spectral line shapes significantly. Two major approaches have been proposed to encounter this problem. The first set of methods that have been demonstrated as reducing the NRB contribution to the measured CARS spectra are optical-based approaches such as polarization CARS [7], frequency modulation CARS [8], single-frequency CARS [9], pulse shaping CARS [10], and interferometric CARS [11]. The second set are computational-based approaches, for example, the maximum entropy method (MEM) [12,13], Kramers-Kronig (KK) relations [14], wavelet prism decomposition analysis [15], phase-error correction [16], and factorized Kramers-Kronig

#530560 https://doi.org/10.1364/OPTCON.530560
Journal © 2024 Received 22 May 2024; revised 10 Jul 2024; accepted 24 Jul 2024; published 6 Aug 2024

error correction [17]. In all these cases, the fixed phase relationship between the Raman and NRB component is adapted to extract the Raman signal from the CARS spectrum.

While all the above-mentioned approaches can reduce the unwanted NRB factor, it is at the cost of experimental complexity and, most importantly, significant signal intensity loss. A state-of-the-art method to overcome the need for these costly techniques is deep learning (DL), which has caused a paradigm change in fields such as computer vision [18], natural language processing [19], and many others [20,21]. Spectroscopic applications such as hyperspectral image analysis [22], vibrational spectroscopy [23,24], and molecular excitation spectroscopy [25] are among the spectroscopy-related fields where DL models could valuably be explored.

Recently, interest has emerged in using DL models to address the NRB issue in CARS measurements. For example, Valensise et al. [26] have proposed a convolutional neural network (CNN) model, named SpecNet, for removing the NRB and extracting the Raman signal from CARS spectra. Wang et al. [27] employed a Very Deep Convolutional Autoencoder (VECTOR) to retrieve the Raman signal from CARS spectra and compared their results with SpecNet that shows improvement in all nine datasets that they have used. In another approach, Houhou et al. [28] utilized a Long Short-Term Memory (LSTM) model as a phase retrieval tool to extract the Raman signal and compared the results with the MEM and KK relation methods which indicated the superiority of the LSTM approach.

In our previous works, we demonstrated the retrieval of Raman signals from CARS spectra by fine-tuning the SpecNet model [29], using different simulated NRBs [30], and deploying a comparison between different DL models including CNN, LSTM, Bidirectional LSTM, and autoencoder models [31]. In this study, we return to the relatively simple SpecNet as the base CNN model and optimize its hyperparameters to investigate if the result of extraction of Raman signals from CARS spectra can be significantly improved by addressing its main drawbacks. The training data are pure synthetic spectra that are similar to the data on which the original SpecNet has been trained.

2. Theory

The four-wave mixing CARS process is comprised of three incident laser fields, namely, pump, Stokes, and probe that are represented as $E_{\mathrm{pu}}(\omega_{\mathrm{pu}})$, $E_{\mathrm{S}}(\omega_{\mathrm{S}})$, and $E_{\mathrm{pr}}(\omega_{\mathrm{pr}})$ respectively. After the broadband Stokes and narrowband pump and probe fields interact coherently with the investigated sample, an anti-Stokes field is produced at the frequency of $\omega_{\mathrm{as}} = \omega_{\mathrm{pu}} - \omega_{\mathrm{S}} + \omega_{\mathrm{pr}}$. When the three fields of pump, Stokes, and probe are parallel polarized, the CARS signal intensity can be donated as follows:

$$I_{\mathrm{CARS}}(\omega_{\mathrm{as}}) \propto \bigg| \iiint_{-\infty}^{\infty} d\omega_{\mathrm{pu}}\, d\omega_{\mathrm{S}}\, d\omega_{\mathrm{pr}}\, \chi^{(3)}_{1111}(\omega_{\mathrm{as}}) \\ E_{\mathrm{pu}}(\omega_{\mathrm{pu}})\, E_{\mathrm{S}}(\omega_{\mathrm{S}})\, E_{\mathrm{pr}}(\omega_{\mathrm{pr}})\, \delta(\omega_{\mathrm{pu}} + \omega_{\mathrm{pr}} - \omega_{\mathrm{S}} - \omega_{\mathrm{as}}) \bigg|^2 \tag{1}$$

where $\delta(\omega_{\mathrm{pu}} + \omega_{\mathrm{pr}} - \omega_{\mathrm{S}} - \omega_{\mathrm{as}})$ is a delta function and $\chi^{(3)}_{1111}$ is a third order tensor that is the sum of $\chi^{(3)}_{\mathrm{NR}}$, a non-resonant, real constant, and frequency independent factor that is raised from the electronic contribution, and $\chi^{(3)}_{\mathrm{R}}$ as described in Eq. 2:

$$\chi^{(3)}_{1111} = \chi^{(3)}_{\mathrm{NR}} + \chi^{(3)}_{\mathrm{R}}(\omega_{\mathrm{as}}). \tag{2}$$

Here $\chi^{(3)}_{\mathrm{R}}$ is a complex vibrational susceptibility that contains chemical information about the sample and can be modeled as:

$$\chi^{(3)}_{\mathrm{R}}(\omega_{\mathrm{as}}) = \sum_{\mathrm{j}} \frac{A_{\mathrm{j}}}{\Omega_{\mathrm{j}} - (\omega_{\mathrm{pu}} - \omega_{\mathrm{S}}) - i\Gamma_{\mathrm{j}}} \tag{3}$$

where for the j^{th} resonance, A_{j}, Ω_{j}, and Γ_{j} are the amplitude, resonance, and the line width respectively.

If we divide the CARS signal from the sample by the CARS signal from a reference sample that has no vibrational resonance alongside the frequency range of the measurement, the resultant CARS line shape will be modeled as follows:

$$S(\omega_{\text{as}}) = \frac{|\chi^{(3)}_{\text{NR}} + \chi^{(3)}_{\text{R}}(\omega_{\text{as}})|^2}{|\chi^{(3)}_{\text{NR,ref}}|^2} = |\chi_{\text{nr}} + \chi_{\text{r}}(\omega_{\text{as}})|^2 = \chi^2_{\text{nr}} + 2\chi_{\text{nr}}Re[\chi_{\text{r}}(\omega_{\text{as}})] + |\chi_{\text{r}}(\omega_{\text{as}})|^2 \quad (4)$$

where $\chi_{\text{nr}} = \frac{\chi^{(3)}_{\text{NR}}}{\chi^{(3)}_{\text{NR,ref}}}$ and $\chi_{\text{r}}(\omega_{\text{as}}) = \frac{\chi^{(3)}_{\text{R}}(\omega_{\text{as}})}{\chi^{(3)}_{\text{NR,ref}}}$ are the normalized background term and the normalized Raman mode. The spectral response of each of the three terms in Eq. 4 and their contribution are depicted in Fig. 1. It is obvious that the mixing between the resonant and non-resonant terms in the second term is the source of dispersity in the CARS signal.

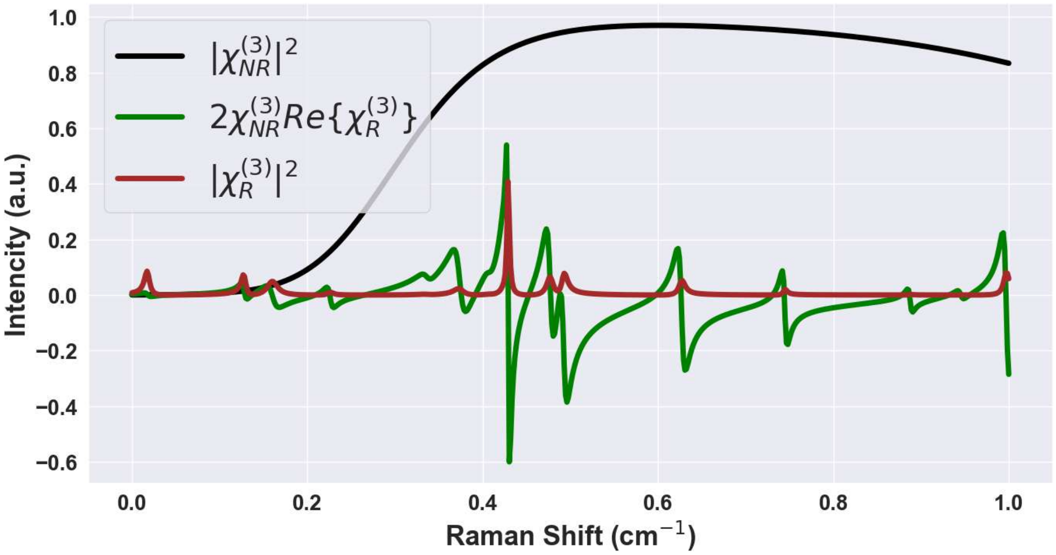

Fig. 1. Spectral response of three terms in CARS signal in the visinity of a single resonance frequency Ω_{R}.

3. Experimental details

3.1. CARS spectra and their corresponding Raman spectra simulation

To simulate the pure synthetic CARS spectra, we used various parameters, including number of peaks, peak amplitude, frequency, linewidth, and a noise factor. The ranges of these parameters are given in Table 1.

The number of peaks per each CARS spectrum is a random number between 1 to 15. Lorentzian peaks, which are sampled by uniform distributions of the amplitude, the resonant frequency, and the linewidth, are as described above. Since the linewidth of the vibrational modes are typically between 1-20 cm^{-1}, the specific range of the linewidth parameter was chosen to cover that distance. We have also added a random noise factor for each of the 640 data points of the spectrum so that the results mimic real CARS spectra. In addition to these parameters, a non-resonant background is added to fulfill the non-resonant part of the CARS spectra. This

Table 1. The parameters of simulated CARS spectra.

S. No	Simulated parameters	Probability distribution
1	No of Peaks	Uniform [1, 15]
2	Peak amplitude (A_i)	Uniform [0.01, 1]
3	Frequency (Ω_i)	Uniform [0, 1]
4	Linewidth (Γ_i)	Uniform [0.001, 0.008]
5	Noise (ϵ_i)	Uniform [0.0005, 0.003]

NRB is a product of two sigmoid functions $\{\sigma_1, \sigma_2\}$ as follows:

$$\mathrm{NRB}(\upsilon) = \sigma_1(\upsilon) * \sigma_2(\upsilon) \tag{5}$$

where the parameters of $\{c_\mathrm{i}, s_\mathrm{i}, \mathrm{i} = 1, 2\}$ are random values that provide a non-uniform background distribution across the spectrum. The main reason to use this NRB is the variability of the produced background shapes that helps increase the model's generalization capability [26].

3.2. Experimental CARS spectra

Four experimental CARS spectra were investigated in this research, namely, an ADP sample, a DMPC lipid, a protein droplet of FUS-LC, and a budding Yeast sample. The optical design of three of the samples is available in [32]. ADP is a mixture of AMP, ADP, and ATP in water with a total concentration of 500 mM, DMPC is a small unilamellar vesicle (SUV) suspension with a concentration of 75 mM, and the living budding Yeast cell sample was measured from the mitochondria of the Yeast cell. The fourth sample is an ultrabroadband CARS spectrum that covers both the fingerprint and the CH-vibration regions of the FUS low complexity domain droplet that was measured by Y. Kan et al. [33].

4. Deep learning model details

4.1. Convolutional neural network (CNN)

A Convolutional Neural Network (CNN) is a type of artificial neural network (ANN) that consists of convolutional, pooling, flattening, and fully connected layers. These models are regularized feed-forward neural networks, which is a unique characteristic, and compared to previous models, they can prevent the vanishing and exploding of gradients in the backpropagation algorithm since the weights are regularized over fewer connections [34]. CNNs have been widely applied in many fields, such as image classification [35], image and video recognition [36], and natural language processing [37]. The convolutional layer, which is a core structure of the CNN model, is performed as the product of the convolution kernel to the current layer nodes. These kernels or filters aim to find low-level features in the first convolutional layers, and the further layers in the architecture learn the more complex features that can properly represent the object. By having different convolutional layers, the model explores more and more detailed features of the input data. The CNN is less dependent on pre-processing of the input data compared to other ANNs since it can learn these features during the model training.

4.2. SpecNet model

The SpecNet model and its architectural parameters are presented in Fig. 2. It is a CNN that is composed of one batch-normalization layer as input, five one-dimensional convolutional layers consisting of (128, 64, 16, 16, 16) nodes and (32, 16, 8, 8, 8) kernel sizes, respectively, two dense layers of 32 and 16 units, a flattening layer, a dropout layer with the dropout rate of 0.25, and finally the output, which is a dense layer of 640 units. The loss function is mean squared error

(MSE), and the optimizer is Adam (adaptive moment estimation) with the default learning rate of 0.001. The batch-normalization layer has four parameters including gamma, beta, moving_mean, and moving_variance. Gamma, a learned scaling factor, and beta, a learned offset factor, are utilized to normalize the output of the layer during the training process. On the other hand, moving_mean and moving_variance are non-trainable parameters that normalize the output during the evaluation. In the figure, conv1d refers to the one-dimensional convolution layer and ReLU represents the rectified linear unit activation function.

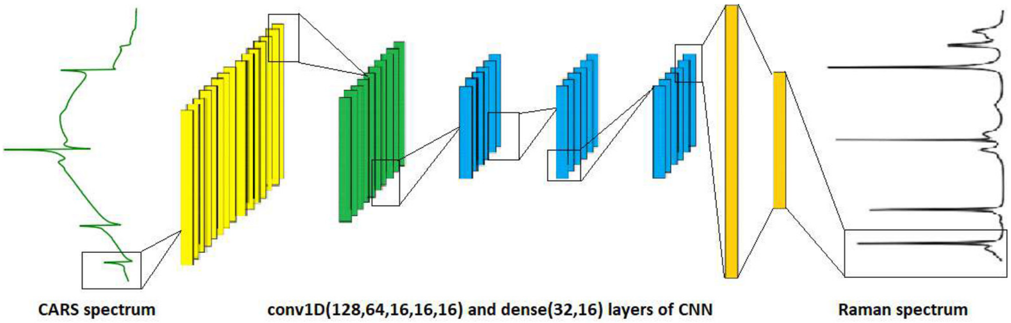

Fig. 2. SpecNet architecture with its architectural parameters, see the text for additional details.

The input to the model is a CARS spectrum with 640 data points, while the output is supposed to be the corresponding Raman signal of same size. The model is trained on pure synthetic data, as described in section 3.1, and performs well on finding the large peaks in the middle of the spectrum. However, SpecNet has two major drawbacks: the first problem is that it cannot find the peaks, large or small, near the edges of the spectral range, and prediction of the Raman signal mostly becomes flat, and the problem of missing some particularly important peaks arises. The second problem is that the model does not recover accurately the intensity of the peaks throughout the spectrum. This is a major drawback because in a problematic case, the concentration of the components of the molecule or the material under investigation would not be properly identified.

5. Results and discussion

To address the two aforementioned problems, we investigated hyperparameter optimization of SpecNet. These hyperparameters are parameters set before training the model, influencing the architecture and the training process, and enhancing the model's performance. The hyperparameters in this CNN architecture include:

- Number of layers including the convolutional, pooling, flattening and fully connected layers.
- Number of nodes per layer.
- Convolution filter size.
- Stride affecting the number of elements that the convolution filter translates at each step.
- Padding affecting how the elements outside the actual data are handled in the convolution and how many elements are affected.
- Activation function introducing the non-linearity.
- L1 and L2 weight regularizers penalizing weights becoming too large.

- Dropout rate representing the probability that a neuron is temporarily excluded from the computation to avoid overfitting.
- Optimizer and the learning rate controlling the updating of the weights.
- Loss function quantifying the difference between the predicted and the desired output.
- Number of epochs limiting the number of iterations in the training process.
- Batch size is the number of samples that are trained per each iteration in each epoch.

5.1. Optimizing the number of nodes and the kernel size of convolutional layers

We explored optimizing the number of neurons and kernel size of the five conv1d layers of the architecture. Initially, we considered ranges for nodes and kernel sizes as (16,. . . , 256) and (3,. . . , 32), respectively. Next, we limited the ranges to [32, 64, 128, 256] for nodes and [8, 16, 32, 64] for kernels. In both cases, we did not enforce the funnel shape of the SpecNet model since each layer could have a random number of nodes.

In contrast to the previous scenario, we kept the funnel shape of the SpecNet model by just optimizing the number of neurons and kernel sizes of the first layer in the limited ranges of [32, 64, 128, 256] and [8, 16, 32, 64], respectively. Then, we divided the resultant optimized numbers of nodes and kernels of the first layer to [2, 8, 8, 8] and [2, 4, 4, 4] for the number of nodes and kernels of the other four convolutional layers.

5.2. Optimizing the L1 and L2 norms, the dropout rate, the learning rate, and the batch size

In SpecNet, L1 and L2 norms, dropout rate, learning rate, and batch size are 0.0, 0.1, 0.25, 0.001, and 256, respectively. We introduced ranges of (0.0, 0.1) for both L1 and L2 regularizers, (0.0, 0.50) for dropout rate, and (1e-5, 1e-2) for the learning rate. Batch size was constrained to [32, 64, 128, 256], which are common values for this hyperparameter.

5.3. Optimizing the padding parameter of convolutional layers

In the convolutional layers, the padding parameter can be one of three options, that is, *valid, same*, or *causal*. If padding is *valid*, it means no padding is added to the layer. On the other hand, padding of *same* will add neurons of zero value to both sides of the layer, so the output has the same dimension as the input. The third option is *causal*, which dilates the convolution filter such that the output [t] does not depend on the input [t+1:].

In SpecNet, the padding was set to default, which is *valid*, and so it did not apply padding through the convolutional layers. In contrast, we changed the padding from *valid* to *same*, and as a result, the number of total parameters of the model increased from 6,016,932 parameters in SpecNet to 6,703,012 parameters.

5.4. Optimizing the loss function

Three loss functions common in regression problems were investigated: mean absolute error (MAE), mean squared error (MSE), and Huber loss. MAE is calculated based on the average of the absolute differences between the predicted and the actual values:

$$\mathrm{MAE} = \frac{1}{n}\sum_{i=1}^{n} |y_i - \widehat{y}_i|. \tag{6}$$

MAE is not very sensitive to outliers, which means a balanced behavior towards all small and large errors. On the other hand, MSE, defined as the average of the squared differences between the predicted and the actual values, that is,

$$\mathrm{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \widehat{y}_i)^2 \tag{7}$$

is more sensitive to outliers. Finally, the Huber loss can be considered as a compromise between MAE and MSE:

$$\mathrm{Huber} = \begin{cases} \frac{1}{2}(y - \widehat{y})^2, & |y - \widehat{y}| \leq \delta \\ \delta \cdot (|y - \widehat{y}| - \frac{1}{2}\delta), & \text{otherwise.} \end{cases} \tag{8}$$

Huber behaves like MAE with smaller errors and like MSE towards the larger errors. The hyperparameter δ introduces a point where the Huber loss converges from quadratic to linear behavior.

In the experiments, we applied an open-source Python library Optuna [38] to optimize the SpecNet model hyperparameters. To implement the optimization, we defined the objective function as the validation accuracy. In two experiments, we run this objective function 100 times in 100 epochs with 30,000 (the original number of samples for training SpecNet) and 100,000 synthetic spectra to optimize the hyperparameters.

To keep the general structure of SpecNet, we have not optimized the number of layers of the model. The stride was kept the same as SpecNet so the convolution filters will cover all neurons in the layers. Also, the activation function of all layers was ReLU.

The two optimized models were named as SpecNet_opt_30 and SpecNet_opt_100. A comparison of their optimized hyperparameters including L1 norm, L2 norm, dropout rate, learning rate, loss function, and batch size with the original SpecNet is shown in Table 2. When comparing the optimized values of the hyperparameters to the original ones, the main observation (in addition to the deliberately changed padding and loss) is that the norm penalties and the dropout rate become much smaller.

Table 2. The hyperparameters of SpecNet, SpecNet_opt_30, and SpecNet_opt_100.

Hyperparameter	SpecNet	SpecNet_opt_30	SpecNet_opt_100
Neurons per conv1d	[128,64,16,16,16]	[128,64,16,16,16]	[128,64,16,16,16]
Kernels per conv1d	[32,16,8,8,8]	[32,16,8,8,8]	[32,16,8,8,8]
Strides of conv1d	1	1	1
Padding of conv1d	“valid”	“same”	“same”
L1, L2 norms of dense layer_1	[0.0, 0.1]	[2.5e-06, 20e-06]	[11e-06, 24e-06]
L1, L2 norms of dense layer_2	[0.0, 0.1]	[3.8e-06, 500e-06]	[110e-06, 37e-06]
Dropout rate of flatten layer	0.25	0.020	0.048
Optimizer/Learning rate	Adam / 0.001	Adam / 0.001	Adam / 0.0005
Loss function	MSE	Huber (delta=3.0)	Huber (delta=1.2)
Batch size	256	32	64

After substituting the optimized hyperparameters to the SpecNet_opt_30 and SpecNet_opt_100 models, we compared the accuracy and loss performance of the three models with each other as shown in Fig. 3. It can be seen that there are notable differences. (1) Overfitting has occurred with the SpecNet model based on both accuracy and loss curves. In both cases, especially in the accuracy curve, the distance between the training and validation plots increases as the process grows. On the other hand, with the optimized models, SpecNet_opt_30 and SpecNet_opt_100, the training and validation curves converge throughout the training process although the convergence is better in the case of SpecNet_opt_100. (2) In the case of the SpecNet model, validation accuracy is higher than training accuracy, while the validation loss is less than the training loss. This is uncommon since it implies that the model performs better with the data that it does not see while training. Potential explanations are: (i) the regularization is applied during training, but not during validation/testing, (ii) the training accuracy/loss is measured during each epoch while validation accuracy/loss is measured after each epoch, and (iii) the validation set may be easier than the training set. (3) The minimum loss of SpecNet is in the order of 1e-3 to 1e-4 while the minimum loss of SpecNet_opt_30 and SpecNet_opt_100 is in the order of 1e-4 and 1e-5 respectively. Also, the accuracy of SpecNet_opt_30 and SpecNet_opt_100 reached approx. 93% and 96% respectively, which shows this model works well while not getting overfitted.

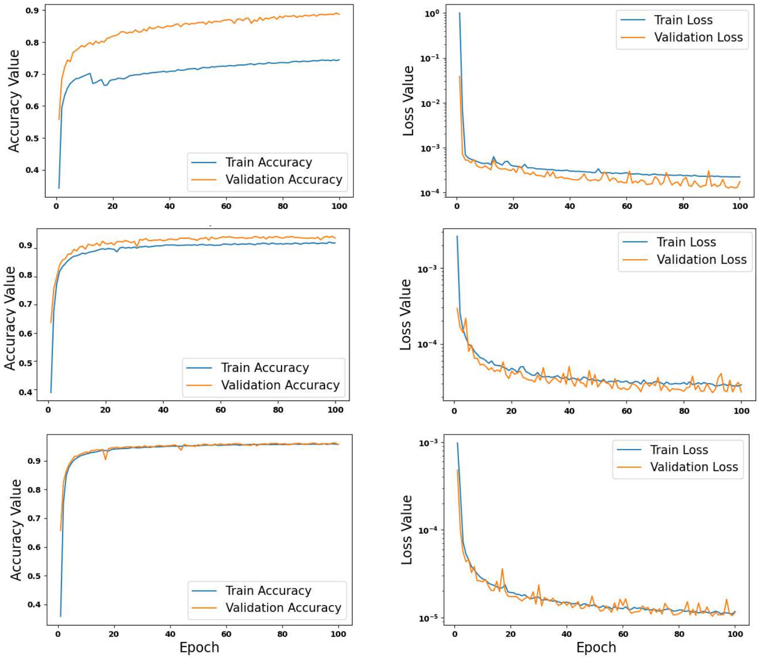

Fig. 3. Comparison of the performance of SpecNet, top, SpecNet_opt_30, middle, and SpecNet_opt_100, bottom, models with regards to both accuracy and loss curves over 100 epochs.

5.5. SpecNet's problem 1: Peaks near the edges of the spectrum

As discussed earlier, the first problem of the SpecNet model when predicting the Raman signal of the input CARS spectrum is that it cannot find the peaks near the edges of the spectrum, and it produces a relatively flat output in those areas. Our strategy to solve this problem was to convert the padding of the convolutional layers from *valid* in SpecNet to *same* in SpecNet_opt. In the context of neural networks, padding *same* means that zeros are added to the outside regions of the input data before applying the convolutional operation, so the output dimension stays the same. In our case, a one-dimensional spectrum, padding *same* will add zeros to both sides of the input spectrum to prevent losing elements after applying the convolutional operation. This is important when the features at the edges of the input spectrum have valuable information and have a crucial role in the prediction task of the model.

5.6. SpecNet's problem 2: Recovery of spectral peak intensities

To solve this problem, we focused on the loss function. The original SpecNet used the MSE function as their loss function. As we have described above, MSE, although a very popular loss function in the context of regression problems, is very sensitive to outliers, which means it will harshly penalize large errors. On the other hand, the MAE loss function does not penalize outliers. Huber loss is a combination of the MAE and the MSE, and while it is sensitive to outliers, it does not penalize them so harshly that it may affect the generality of the model. This characteristic helped to address the intensity problem mostly in the middle range of the spectrum, where on some occasions the SpecNet model produces a flat output and cannot recover either the high peaks or the small peaks.

To summarize, using the padding *same* in convolutional layers, Huber loss as the loss function, and other optimized hyperparameters including L1, L2, dropout rate, learning rate, and the number of batch size, helped to improve the prediction accuracy of the optimized models SpecNet_opt_30 and SpecNet_opt_100 compared to the original SpecNet.

5.7. Prediction based on synthetic CARS spectra

The performance of the three models in prediction of 1000 unseen synthetic CARS spectra by calculating their mean squared errors (MSE), standard deviations (StD), and medinas with the actual targets is presented in Table 3. The average error of both SpecNet_opt_30 and SpecNet_opt_100 model is approx. 10^{-5} while the average error of the SpecNet model is at 10^{-3}. The boxplot, Fig. 4, visually shows the distribution of errors including medians, quartiles, whiskers and outliers of the investigated samples. The median for SpecNet is approx. 10^{-3} while it is around 10^{-5} for both SpecNet_opt_30 and SpecNet_opt_100 that suggests a significant improvement in performance by reducing the central error.

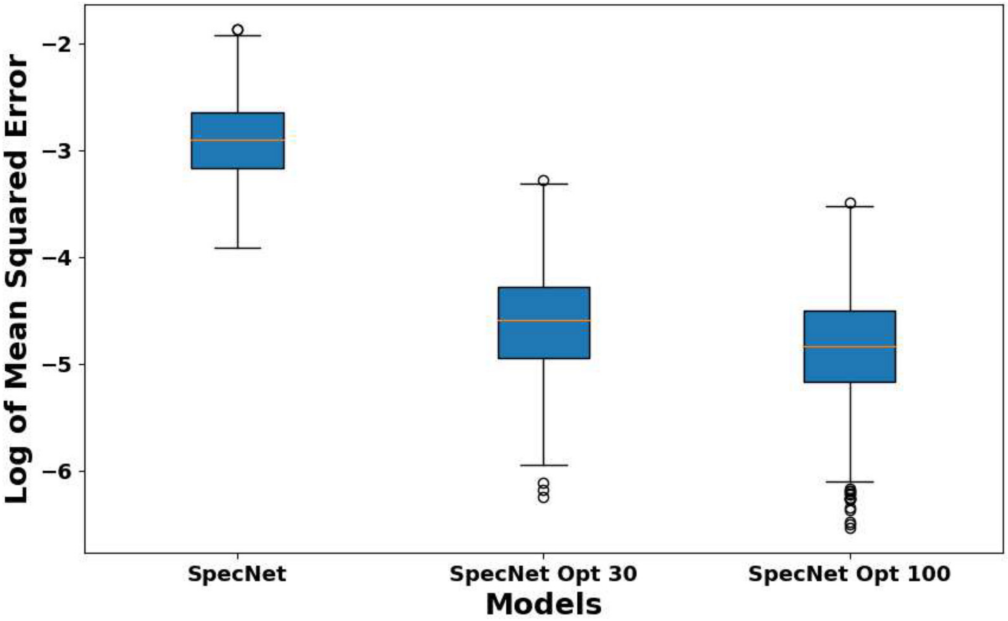

Fig. 4. Distribution of errors including medians, quartiles and outliers of three models using a boxplot.

These results is not surprising since the accuracy of the optimized models of SpecNet_opt_30 and SpecNet_opt_100 is approx. 93% and 96%. Based on the results of the mean of MSEs/StDs,

Table 3. Mean of MSEs, StDs and medians of three models.

Model	Mean MSE	Mean StD	Median
SpecNet	1.719×10^{-3}	1.601×10^{-3}	-2.909833
SpecNet_opt_30	3.9×10^{-5}	4.6×10^{-5}	-4.594653
SpecNet_opt_100	2.4×10^{-5}	3.0×10^{-5}	-4.836616

and medinas of three models in Table 3, the accuracy-loss analysis from Fig. 3, the results of predicting the synthetic samples, Fig. 4, and the results from the experimental spectra, given in Supplement 1, and since the SpecNet_opt_100 model works better than the SpecNet_opt_30 model, henceforth we will take SpecNet_opt_100 as the optimized model and will continue with this model.

In Fig. 5, we present the prediction of the Raman signals for two synthetic CARS spectra by SpecNet and SpecNet_opt_100 and compare them with their corresponding true Raman signals. Additionally, the squared errors (SEs) of both models were compared with their true Raman signals and they are presented as well. These two spectra were selected since they are the worst (173rd spectrum in the test set) and the best (815th spectrum in the test set) predictions by SpecNet. In the top figure, representing the worst case, the SpecNet model has failed to find three

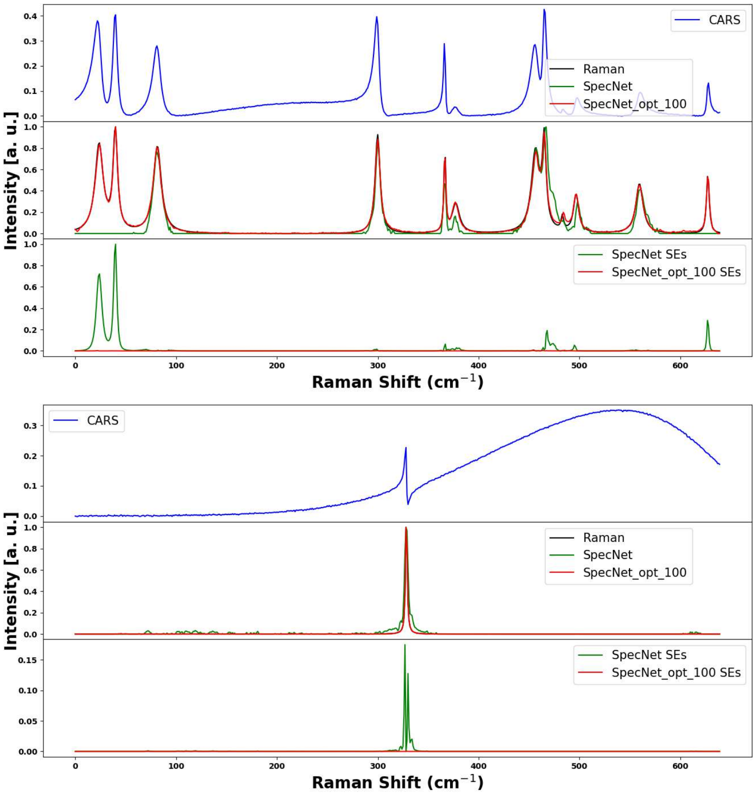

Fig. 5. Prediction of the true Raman signals (black) by SpecNet (green) and optimized model of SpecNet_opt_100 (red) for two synthetic CARS spectra. The corresponding squared errors of these two signals (173, 815) were also calculated and are presented in the figure.

OPTICS CONTINUUM

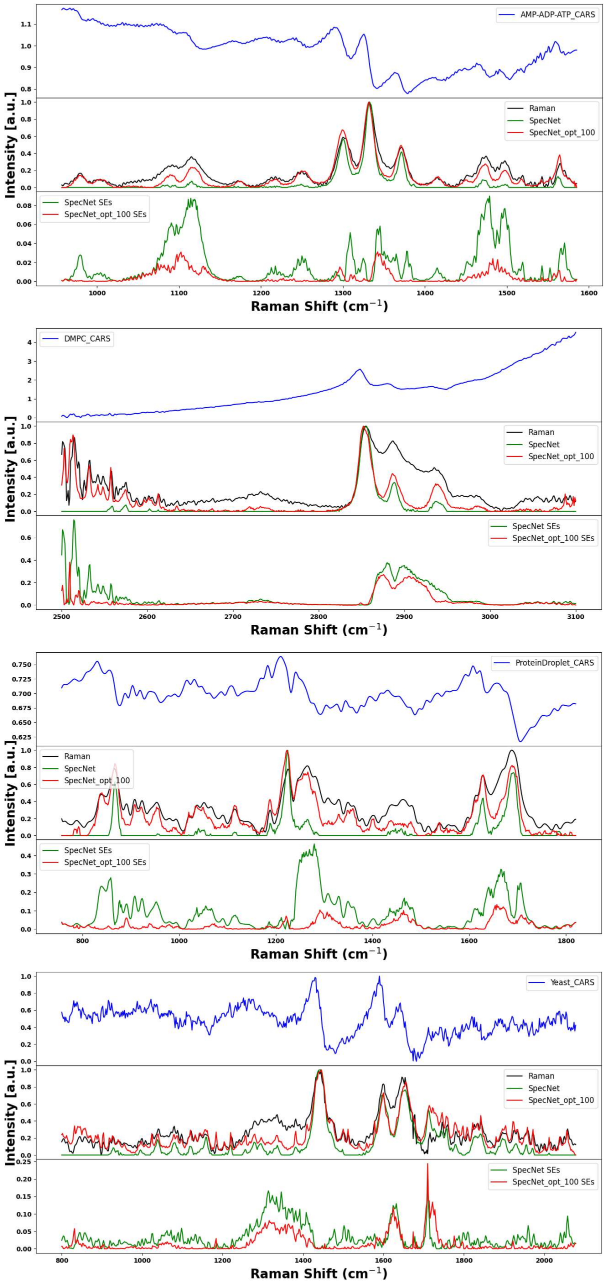

Fig. 6. Prediction of the true Raman signals (black) by SpecNet (green) and SpecNet_opt_100 (red) and the corresponding squared errors of four experimental CARS spectra for an AMP/ADP/ATP mixture, DMPC, Protein Droplet, and Yeast samples.

relatively high peaks at both edges. Furthermore, due to the level of noise in the input CARS spectrum, the predicted signal also shows a noisy result. In contrast, SpecNet_opt_100 was able to predict all three peaks near the edges and the intensity of all peaks along the spectrum and the noise in the CARS spectrum did not distort its prediction. The results of the second synthetic example, bottom figure, show that although the level of noise is more than in the previous CARS signal, SpecNet_opt_100 was able to identify the main peak in the middle of the spectrum and its intensity, while the SpecNet model was distorted by the noise, especially on the right side of the spectrum, producing many small artefacts and lost the correct intensity of the peak as well.

5.8. Prediction based on experimental CARS spectra

This section examines the performance of the two models, SpecNet and SpecNet_opt_100, in retrieving the Raman signal from experimentally measured CARS spectra. In addition, the squared errors of the models compared to the corresponding true Raman spectra are provided. The four real CARS spectra, which were not used in the training process, were an ADP/AMP/ATP mixture, a DMPC lipid, a protein Droplet, and a Yeast sample. Their preparation details and an experimental overview were presented in section 4.2.

These biological samples have different resonance vibrational bands, various backgrounds, and noise. The mixture sample in Fig. 6 was predicted accurately by the SpecNet_opt_100 model, since it predicted the left side small peaks of the Raman line shape and also the intensity of the peaks, especially the major peaks in the middle of the signal and the right side of the spectrum, whereas SpecNet has become totally flat in that area before the Raman shift of 1100. In the case of DMPC, SpecNet_opt_100 has clearly identified the relatively pronounced peaks at the left edge of the spectrum, whereas the SpecNet prediction is almost flat. On the right side of the spectrum, from the Raman shift of 2980 until the end of the spectral range, SpecNet is flat, while SpecNet_opt_100 has recovered some small peaks. With the droplet protein sample, SpecNet not only produces a flat output at both ends of the spectrum, but the output has become totally flat in three parts of the middle of the spectrum, and it has consequently failed to find these relatively large peaks. SpecNet_opt_100 performs well. The Yeast sample is noisier than the other samples. Nevertheless, SpecNet_opt_100 can almost accurately predict the peaks at both ends of the spectrum, while the SpecNet model produces a flat output at those areas. In addition, the SpecNet_opt_100 model has matched the intensity of the true Raman signal in the major and narrow bands throughout the spectrum.

6. Conclusion

In this work, we optimized the SpecNet model by using the Optuna Python library to fine-tune the model's hyperparameters. These parameters include the number of nodes and kernels in the convolutional layers, padding in the convolution layers, L1 and L2 norms of the dense layers, optimizer learning rate, dropout rate for the flatten layer, loss function, and batch size for training the model. The objective function selected for the optimization was validation accuracy of the model, and 30,000 and 100,000 synthetic spectra were used for the process.

Three models, the original SpecNet and two optimized ones, SpecNet_opt_30 and SpecNet_opt_100, were trained on the same set of synthetic spectra. The models were compared by first predicting the Raman signal from new synthetic CARS data and determining the mean squared errors of the models. Next, we explored the models' performance on four real biological CARS spectra, namely, an ADP/AMP/ATP mixture, a DMPC sample, a droplet protein sample, and a Yeast sample. The experiments showed that the optimized SpecNet_opt_100 model is clearly superior to the original SpecNet and the other optimized model, SpecNet_opt_30, at finding the spectral peaks at both ends of the spectrum and recovering the intensity of the spectral peaks. The results demonstrate the importance of hyperparameter optimization and show that a CNN is a viable option for recovering Raman signals from CARS spectra and show that a CNN is a viable option for recovering Raman signals from CARS spectra.

Funding. Research Council of Finland (grant no. FIRI/327734).

Acknowledgments. This work continues the "Quantitative Chemically-Specific Imaging Infrastructure for Material and Life Sciences (qCSI)" project funded by the Academy of Finland (grant no. FIRI/327734). Also, we thank Michiel Müller and Hilde Rinia for providing the experimental measurements of the DMPC lipid sample and the AMP/ADP/ATP

OPTICS CONTINUUM

mixture, Masanari Okuno and Hideaki Kano for providing the experimental measurements of the yeast sample, and Yelena Kan and Sapun Parekh for providing the protein droplet measurements.

Disclosures. The authors declare no conflicts of interest.

Data availability. The data supporting this study's findings are available from the corresponding author on request.

Supplemental document. See Supplement 1 for supporting content.

References

1. A. Zumbusch, G. R. Holtom, and X. S. Xie, "Three-dimensional vibrational imaging by coherent anti-Stokes Raman scattering," Phys. Rev. Lett. **82**(20), 4142–4145 (1999).
2. J.-X. Cheng, L. D. Book, and X. S. Xie, "Polarization coherent anti-Stokes Raman scattering microscopy," Opt. Lett. **26**(17), 1341–1343 (2001).
3. L. M. Malard, L. Lafeta, R. S. Cunha, *et al.*, "Studying 2d materials with advanced Raman spectroscopy: CARS, SRS and TERS," Phys. Chem. Chem. Phys. **23**(41), 23428–23444 (2021).
4. M. Tamamitsu, Y. Sakaki, T. Nakamura, *et al.*, "Ultrafast broadband Fourier-transform CARS spectroscopy at 50,000 spectra/s enabled by a scanning Fourier-domain delay line," Vib. Spectrosc. **91**, 163–169 (2017).
5. F. Ganikhanov, C. L. Evans, B. G. Saar, *et al.*, "High-sensitivity vibrational imaging with frequency modulation coherent anti-Stokes Raman scattering (FM CARS) microscopy," Opt. Lett. **31**(12), 1872–1874 (2006).
6. G. I. Petrov, R. Arora, and V. V. Yakovlev, "Coherent anti-Stokes Raman scattering imaging of microcalcifications associated with breast cancer," Analyst **146**(4), 1253–1259 (2021).
7. C. L. Evans, E. O. Potma, M. Puoris' haag, *et al.*, "Chemical imaging of tissue in vivo with video-rate coherent anti-Stokes Raman scattering microscopy," Proc. Natl. Acad. Sci. U.S.A. **102**(46), 16807–16812 (2005).
8. C. L. Evans and X. S. Xie, "Coherent anti-Stokes Raman scattering microscopy: chemical imaging for biology and medicine," Annual Rev. Anal. Chem. **1**, 883–909 (2008).
9. O. Burkacky, A. Zumbusch, C. Brackmann, *et al.*, "Dual-pump coherent anti-Stokes-Raman scattering microscopy," Opt. Lett. **31**(24), 3656–3658 (2006).
10. S. O. Konorov, M. W. Blades, and R. F. Turner, "Lorentzian amplitude and phase pulse shaping for nonresonant background suppression and enhanced spectral resolution in coherent anti-Stokes Raman scattering spectroscopy and microscopy," Appl. Spectrosc. **64**(7), 767–774 (2010).
11. M. Jurna, J. Korterik, C. Otto, *et al.*, "Background free CARS imaging by phase sensitive heterodyne CARS," Opt. Express **16**(20), 15863–15869 (2008).
12. E. M. Vartiainen, "Phase retrieval approach for coherent anti-Stokes Raman scattering spectrum analysis," J. Opt. Soc. Am. B **9**(8), 1209–1214 (1992).
13. E. M. Vartiainen, H. A. Rinia, M. Müller, *et al.*, "Direct extraction of Raman line-shapes from congested CARS spectra," Opt. Express **14**(8), 3622–3630 (2006).
14. Y. Liu, Y. J. Lee, and M. T. Cicerone, "Broadband CARS spectral phase retrieval using a time-domain Kramers–Kronig transform," Opt. Lett. **34**(9), 1363–1365 (2009).
15. Y. Kan, L. Lensu, G. Hehl, *et al.*, "Wavelet prism decomposition analysis applied to CARS spectroscopy: a tool for accurate and quantitative extraction of resonant vibrational responses," Opt. Express **24**(11), 11905–11916 (2016).
16. C. H. Camp Jr, Y. J. Lee, and M. T. Cicerone, "Quantitative, comparable coherent anti-Stokes Raman scattering (CARS) spectroscopy: correcting errors in phase retrieval," J. Raman Spectrosc. **47**, 408–415 (2016).
17. C. H. Camp Jr, J. S. Bender, and Y. J. Lee, "Real-time and high-throughput Raman signal extraction and processing in CARS hyperspectral imaging," Opt. Express **28**(14), 20422–20437 (2020).
18. Y. Guo, Y. Liu, A. Oerlemans, *et al.*, "Deep learning for visual understanding: A review," Neurocomputing **187**, 27–48 (2016).
19. D. W. Otter, J. R. Medina, and J. K. Kalita, "A survey of the usages of deep learning for natural language processing," IEEE Trans. Neural Netw. Learning Syst. **32**(2), 604–624 (2021).
20. H. Liang, X. Sun, Y. Sun, *et al.*, "Text feature extraction based on deep learning: a review," EURASIP Journal on Wireless Communications and Networking **2017**(1), 211 (2017).
21. J. Du, Y. Liu, and Z. Liu, "Study of precipitation forecast based on deep belief networks," Algorithms **11**(9), 132 (2018).
22. F. Lussier, V. Thibault, B. Charron, *et al.*, "Deep learning and artificial intelligence methods for Raman and surface-enhanced Raman scattering," TrAC, Trends Anal. Chem. **124**, 115796 (2020).
23. J. Yang, J. Xu, X. Zhang, *et al.*, "Deep learning for vibrational spectral analysis: Recent progress and a practical guide," Anal. Chim. Acta **1081**, 6–17 (2019).
24. R. Luo, J. Popp, and T. Bocklitz, "Deep learning for Raman spectroscopy: A review," Analytica **3**(3), 287–301 (2022).
25. K. Ghosh, A. Stuke, M. Todorović, *et al.*, "Deep learning spectroscopy: Neural networks for molecular excitation spectra," Advanced Science **6**, 1801367 (2019).
26. C. M. Valensise, A. Giuseppi, F. Vernuccio, *et al.*, "Removing non-resonant background from CARS spectra via deep learning," APL Photonics **5**(6), 061305 (2020).

27. Z. Wang, K. O'Dwyer, R. Muddiman, *et al.*, "VECTOR: Very deep convolutional autoencoders for non-resonant background removal in broadband coherent anti-Stokes Raman scattering," J. Raman Spectrosc. **53**, 1081–1093 (2022).
28. R. Houhou, P. Barman, M. Schmitt, *et al.*, "Deep learning as phase retrieval tool for CARS spectra," Opt. Express **28**(14), 21002–21024 (2020).
29. A. Saghi, R. Junjuri, L. Lensu, *et al.*, "Semi-synthetic data generation to fine-tune a convolutional neural network for retrieving Raman signals from CARS spectra," Opt. Continuum **1**(11), 2360–2373 (2022).
30. R. Junjuri, A. Saghi, L. Lensu, *et al.*, "Effect of non-resonant background on the extraction of Raman signals from CARS spectra using deep neural networks," RSC Adv. **12**(44), 28755–28766 (2022).
31. R. Junjuri, A. Saghi, L. Lensu, *et al.*, "Evaluating different deep learning models for efficient extraction of Raman signals from CARS spectra," Phys. Chem. Chem. Phys. **25**(24), 16340–16353 (2023).
32. M. Müller and J. M. Schins, "Imaging the thermodynamic state of lipid membranes with multiplex CARS microscopy," J. Phys. Chem. B **106**(14), 3715–3723 (2002).
33. A. C. Murthy, G. L. Dignon, Y. Kan, *et al.*, "Molecular interactions underlying liquid-liquid phase separation of the FUS low-complexity domain," Nat. Struct. Mol. Biol. **26**(7), 637–648 (2019).
34. R. Venkatesan and B. Li, *Convolutional neural networks in visual computing: a concise guide* (CRC Press, 2017).
35. N. Sharma, V. Jain, and A. Mishra, "An analysis of convolutional neural networks for image classification," Procedia Computer Science **132**, 377–384 (2018).
36. M. V. Valueva, N. Nagornov, P. A. Lyakhov, *et al.*, "Application of the residue number system to reduce hardware costs of the convolutional neural network implementation," Mathematics and Computers in Simulation **177**, 232–243 (2020).
37. N. I. Widiastuti, "Convolution neural network for text mining and natural language processing," in *IOP Conference Series: Materials Science and Engineering*, vol. 662 (IOP Publishing, 2019), p. 052010.
38. T. Akiba, S. Sano, T. Yanase, *et al.*, "Optuna: A next-generation hyperparameter optimization framework," in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge & Discovery Data Mining*, (2019), pp. 2623–2631.