Master Thesis

Study on Imaging Processing Methods for Stimulated Raman Scattering Microscopy

誘導ラマン散乱顕微法のための画像処理法の研究

Submitted on Aug. 14, 2019
Advisor: Associate Professor OZEKI, Yasuyuki

Department of Electrical Engineering and Information Systems
Graduate School of Engineering, The University of Tokyo

Student Number: 37-175081
LIU, Hanqin

Table of Contents

Acknowledgment

I would like to especially thank Dr. Yasuyuki Ozeki for his instructions and encouragements throughout the two years. Discussions with him were always idea-provoking and encouraging. He suggested me to evaluate the current spectral unmixing algorithms and that directly led to the findings in Chapter 4 and Chapter 6. He also gave me valuable advice concerning the CNN part—with his guidance, the classification of nucleolus got much improved. I would be much at sea without his great help.

I also appreciate Dr. Yuta Suzuki to a great degree. He involved me into the imaging analysis part of his project, where I deepened the wholistic understanding of spectral unmixing. He also gave me a lot of critical questions about my research and pushed me to dig deeper and deeper.

Thanks also go to Mr. Takuya Asai, who provided me with almost all the SRS images used in this thesis. It would be quite hard to imagine how to progress on imaging analysis if there were no raw images to be analyzed. With his ingenious and careful experiment design, I could get SRS images with high quality for further analysis.

I am much grateful to Mr. Jingwen Shou too. He helped me familiarize the whole SRS system with his profound understanding of optics. He is now working on combining fluorescence imaging and SRS imaging, which I believe would be powerful and would reveal us abundant information about cells. It was he who provided me the fluorescent images of the nucleus in Chapter 5, which served as a key to reducing the huge training cost. Besides the research, he also helped me settle down in Japan at the very beginning and helped me fit into livings here in a short time.

I am also much appreciated for what Mr. Chun-Jung Huang has taught me. He impressed me with his expertise in CNN and provided me with a lot of useful techniques in modifying a CNN network. His work in CNN intrigued me a lot and finally made me incorporate CNN architecture into my work as well. I am thankful to Dr. Choon Pin Foong from Riken institute too. He provided me with PHA bacteria for the interesting work listed in the appendix

1. Introduction

Stimulated Raman scattering (SRS) is one kind of label-free, non-invasive bioimaging technique that provides us molecular vibrational signatures of biological samples. This thesis mainly focuses on how to analyze SRS images to retrieve useful information. There is two key information that we are interested in: the molecular vibrational signatures (spectrum) and the concentration distribution of this molecular. The process of recovering the two information from the raw SRS images is called "spectral unmixing" or "decomposition." I will use the two terms interchangeably in the following chapters.

Although there were a lot of algorithms that were used for spectral unmixing, their strength and weakness had not been clearly investigated. Therefore, I first comprehensively researched the existing methods and pointed out their respective advantages and disadvantages. I also compared and evaluated their performance under noisy situation and found that the performance of different algorithms is component-dependent, i.e., performance varies when different algorithms are used in unmixing the same component if the signal-to-noise ratio is low. This finding would not only shed light on the logic behind the algorithms but also give us instruction on how to choose a proper method under a certain situation.

Furthermore, I also trained a convolutional neural network (CNN) to enable a fast decomposition. In this way, not only spectral information but also spatial information can be taken into consideration during decomposition. It turned out that a pre-trained CNN would outperform the state-of-art unmixing algorithms given a limited number of spectral channels. Meanwhile, I also applied CNN in segmenting cell structures, e.g., cytoplasm and nucleus, and it worked reasonably well.

The decomposition aforementioned mainly refers to unmixing components that have significant differences in the vibrational spectrum, e.g., lipids and protein. However, even within one category, there could still be spectral nuances. For example, unsaturated lipids have a higher peak than the saturated lipids around 3010cm^{-1}, but all the rest of the two spectra look similar to each other. I also explored algorithms that can find this nuance, thus enable us to have a more detailed perception of the concentration map for components with minor spectral differences.

The thesis is organized as follows: Chapter 2 gives a brief introduction of the principle of SRS microscopy and experimental setup. Chapter 3 focuses on spectral unmixing, gives the mathematical definition of the problem, and introduces existing algorithms. Chapter 4 critically evaluates the existing methods and compares their performance under noise; chapter 5 introduces a CNN architecture known as U-net into the spectral unmixing and the segmentation of cell structures. Chapter 6 emphasizes on the discrimination of finer spectral differences.

2. Stimulated Raman scattering (SRS) microscopy

At present, most biomedical imaging techniques rely on staining or labeling to make transparent samples visible under microscopes. These kinds of technologies are time-consuming and thus cannot be applied in the cases that require real-time imaging like surgery. Furthermore, labeling or staining may disturb the normal functioning of cells[1].

As a promising alternative, stimulated Raman scattering (SRS) microscopy is one kind of fast, stain-free bioimaging technology. It exploits SRS effect, which means when the vibrational frequency of molecules matches with the beat frequency of two laser pulses, energy will transfer from the one with higher frequency to the other with lower frequency, and the transferred energy will be proportional to the concentration of molecules of interest.

As shown in the Fig. 1, SRS utilizes two-color laser pulse trains, one of which is intensity-modulated so that when the SRS effect occurs, the intensity modulation will be transferred to the other pulse train. After the two pulses are combined, focused, and then pass through the sample, the transferred modulation is extracted with lock-in detection circuits at the frequency of modulation. In view that the intensity of the transferred energy is proportional to the concentration of molecules of interest, the magnitude of the output signal of the lock-in amplifier will indicate the density of molecules at the current focus point.

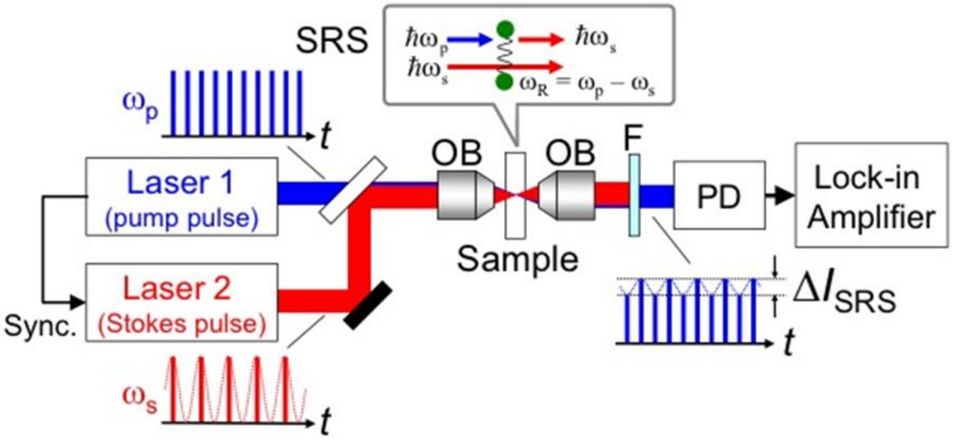


Fig. 1. Principle of SRS. OB: objective lens; PD: photodiode[2]

Considering that the vibration of molecules can serve as a natural identity for biomolecules, by measuring the transferred energy, one can acquire molecular vibrational spectra and then identify different types of biomolecules. Therefore, by utilizing the vibration frequency of molecules as the contrast of imaging, SRS avoids time-consuming labeling process and enable the bioimaging in a fast and label-free manner.

As shown in Fig. 2, in our experimental setup, we used Yb fiber laser and Ti: sapphire laser as two laser sources. The two pulses are controlled by X-Y scanners so that they can be focused at different positions on the sample. Also, Yb fiber laser pulses are tunable, which leads to a changing beat frequency of the two beams, so that one can obtain the vibrational spectrum over a continuous frequency band. The vibrational spectrum serves as the very identity for different components. Finally, a hyperspectral data cube along spatial and spectral dimensions is obtained. This system achieved a frame-by-frame wavelength tunability and reached an imaging speed of 30 frames/s.

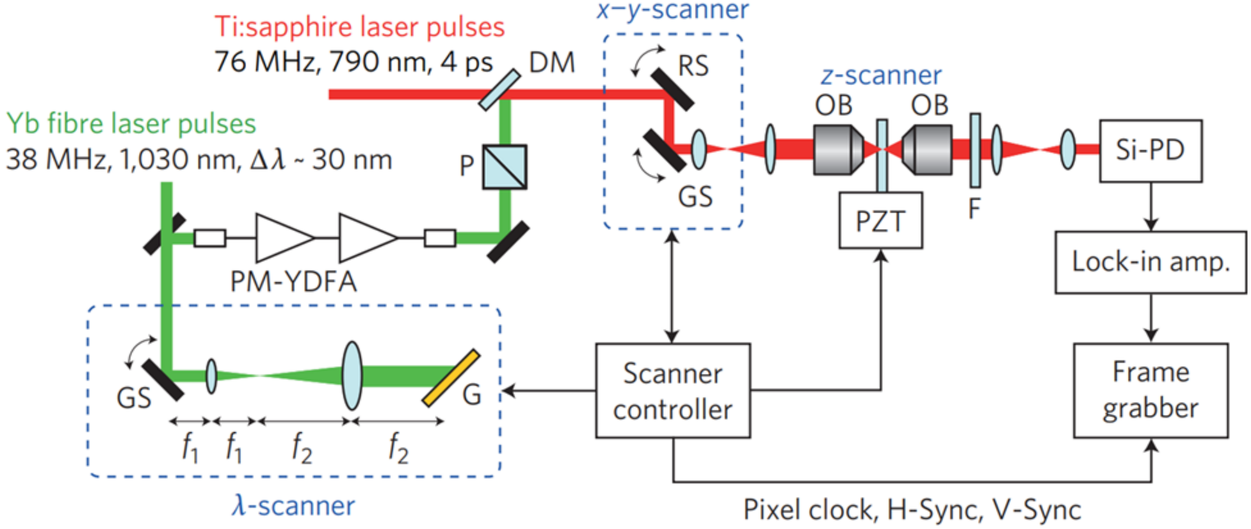


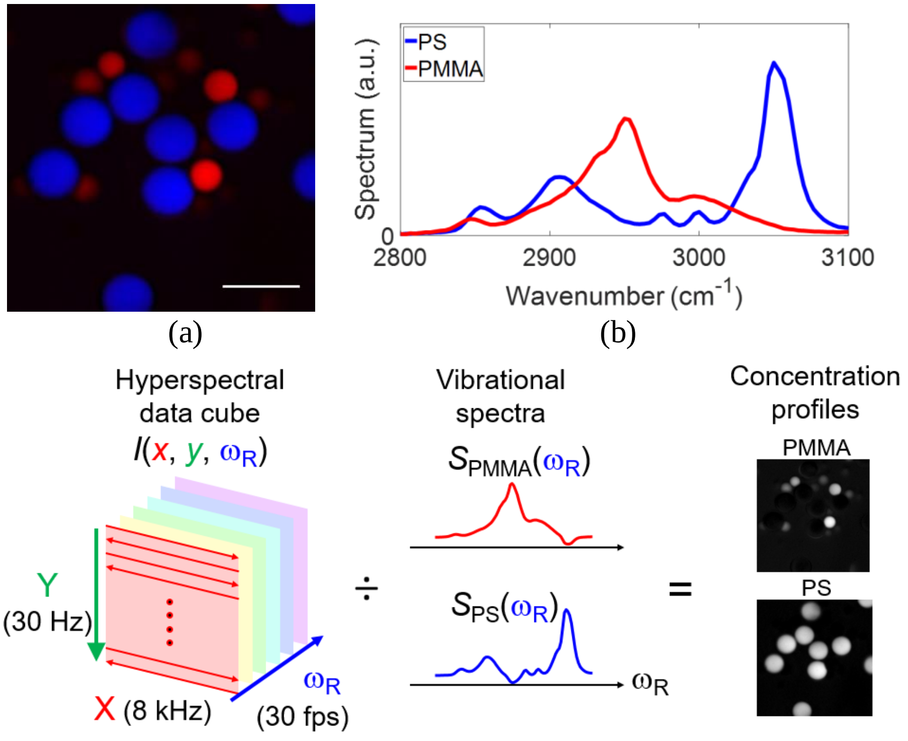
Fig. 2. Experimental setup[3]. PM-YDFA: polarization-maintaining Yb-doped fiber amplifier; GS: galvanometer scanner; G: grating, RS: resonant galvanometer scanner; P: polarizer; DM: dichroic mirror; OB: objective lens; PZT: piezoelectric transducer; F: short-pass filter; f1 = 50 mm, f2 = 100 mm.

3. Previous SRS spectral unmixing methods

In SRS, spectral unmixing is essential to extract meaningful information, e.g., the vibrational spectrum that serves as the identity of molecules. To date, several methods have been applied in spectral unmixing of the SRS data, but their pros and cons have not been explicitly explored. This chapter first introduces the mathematical definition of the spectral unmixing problem and then examines several typical spectral unmixing methods.

3.1. Mathematical definition of spectral unmixing problem

In SRS, there are two kinds of information that are of great interest: the vibrational spectra and the concentration profile of molecules. These two types of information can be recovered from raw SRS data cube following certain steps (see Fig. 3.). First, we examine the SRS images along the axis of wavelength and locate the different components. Then extract the vibrational spectra from the different components. Since the pixel intensity is proportional to the density components, by making a "division" operation, one can further retrieve the concentration profiles.



(a) (b)

(c)

Fig. 3. Unmixing of SRS data. (a) Image obtained by SRS. Two different beads (PS and PMMA) are merged with blue and red. (b) Manually extracted vibrational spectra for

two types of beads. (c) Illustration of the way to get concentration profiles. Scalebar: 20μm

For more rigorous explanation, assume that there are $\boldsymbol{p}$ kinds of components, whose vibrational spectra are $\boldsymbol{S} = [\boldsymbol{s}_1, \boldsymbol{s}_2, \ldots, \boldsymbol{s}_p]$, where $\boldsymbol{s}_i$ denotes the vibrational spectra of the $\boldsymbol{i}^{th}$ type of molecule. If the number of tunable wavelengths is $\boldsymbol{L}$, then $\boldsymbol{s}_i$ would be an $\boldsymbol{L}$-dimensional column vector. On the other hand, the concentration profile can be denoted as $\boldsymbol{C} = [\boldsymbol{c}_1, \boldsymbol{c}_2, \ldots, \boldsymbol{c}_p]$, where $\boldsymbol{c}_i$ is a $\boldsymbol{K}$-dimensional column vector that represents the pixelwise concentration profile of the $\boldsymbol{i}^{th}$ component ($\boldsymbol{K}$ is the number of pixels in one image, i.e., if the image is m-by-n, then $\boldsymbol{K} = \boldsymbol{m} \times \boldsymbol{n}$). In this way the original 3-dimensional dataset can be unfolded along the x-y plane, ends up as a 2-dimensional dataset $\boldsymbol{R}$ and it can be modelled as:

$$\boldsymbol{R} = \boldsymbol{S}\boldsymbol{C}' + \boldsymbol{E}, \qquad (1)$$

where $\boldsymbol{E}$ stands for the additive noise. We can represent the equation above in the form of a matrix:

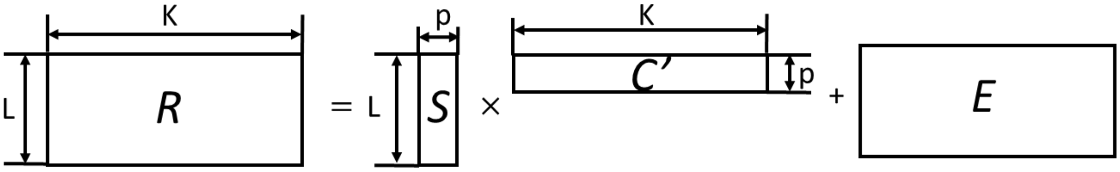


Fig. 4. The matrix form of Eq. 1

The aim of the spectral unmixing is to retrieve the vibrational spectrum $\boldsymbol{S}$ and concentration profile $\boldsymbol{C}$ for the $\boldsymbol{p}$ kinds of molecules from SRS data $\boldsymbol{R}$. One needs to find a way to decompose the dataset $\boldsymbol{R}$ into two smaller matrices that can be hopefully explained as $\boldsymbol{S}$ and $\boldsymbol{C}$.

It is granted that one can unmix the dataset manually, as shown in Fig. 3, the workload, however, will surge if there is a large amount of data. Plus, the pixels selected manually are subject to artifacts and limited numbers. Therefore, it is worthy of finding methods that can unmix the dataset automatically.

Extensive work has been done in trying to retrieve the vibrational spectrum $\boldsymbol{S}$ and concentration profile $\boldsymbol{C}$ from the linearly mixed data $\boldsymbol{R}$. Multivariate curve resolution (MCR) is used to seek approximations of $\boldsymbol{S}$ and $\boldsymbol{C}$ under specific physical constraints, e.g., non-negativity. Principal component analysis (PCA) and Independent component analysis (ICA) exploit different statistical properties of the data, i.e., irrelevance and independence. Vertex component analysis (VCA), however, utilizes the geographic distribution of data to decompose the mixed data. In the following section, the basics of these methods are briefly introduced.

3.2. Spectral unmixing methods

3.2.1. Multivariate curve resolution (MCR)

Intuitively speaking, from Fig. 4 one can find that spectral unmixing can be achieved by finding two smaller matrices, $\boldsymbol{X}$ ($\boldsymbol{L}$ by p) and $\boldsymbol{Y}$ ($\boldsymbol{K}$ by $\boldsymbol{p}$) to approximate $\boldsymbol{S}$ and $\boldsymbol{C}$. In this case, the product of $\boldsymbol{X}$ and $\boldsymbol{Y}$ should be as close as to $\boldsymbol{R}$. The reconstruction error between $\boldsymbol{R}$ and $\boldsymbol{XY}$ can be hopefully explained as $\boldsymbol{E}$. This is the very idea of MCR.

However, Eq.1 is ill-posed, which means that it is impossible to get a single solution for $\boldsymbol{S}$ and $\boldsymbol{C}$ by solving it. To retrieve $\boldsymbol{S}$ and $\boldsymbol{C}$, additional constraints must be imposed to narrow down the space of solutions. Commonly used constraints are shown in the Fig. 5.

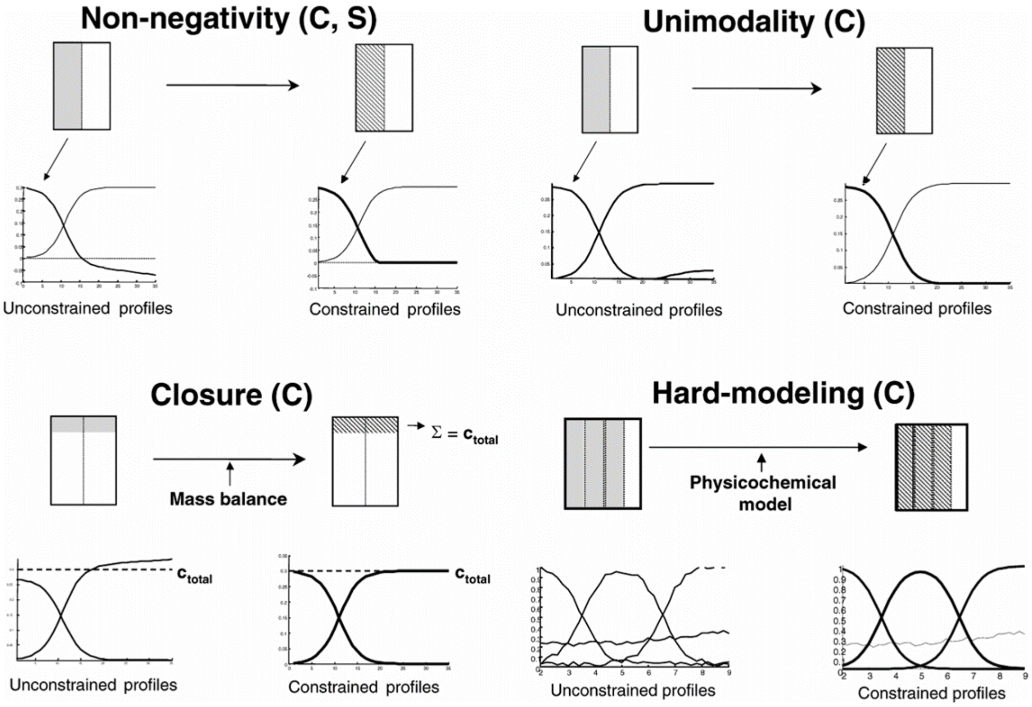


Fig. 5. Common constraints in MCR approach[4]

Non-negativity is the most frequently used constraint. It requires that solutions for $\boldsymbol{C}$ and $\boldsymbol{S}$ should be positive. Also, unimodality is applied when there is only one local maximum value in the concentration profile. Closure requires that the sum of the concentration of two components should be a constant. Hard modeling, however, exploits other physical constraints, e.g., smoothness of components profile. After imposing all the applicable restrictions, MCR will begin to seek solutions that minimize the reconstruction error under all the imposed conditions.

It is worthy to note that, usually the physical constraints above are not strong enough to give a unique and satisfying solution. Therefore, information more than physical constraints should be exploited to achieve better performance.

3.2.2. Principal component analysis (PCA)

PCA is extensively applied for dimension reduction and data reconstruction. It projects the original dataset to a direction that explains the most significant part of the variance, and then to the directions orthogonal to the previous directions. In this way, PCA rotates the data points and reserve all the information[5].

More specifically, the original variables (denoted by the column vectors of ***R***) are decorrelated and recombined into several new variables (indicated by the row vectors of ***P***), so that the new variables account for the main variance of the raw data. These new variables are termed as principal components. The vectors that project ***R*** into the space of principal components are called loading vectors (denoted as ***T***). ***T*** and ***P*** can be attained with the eigendecomposition of the covariance matrix of ***R***: the eigenvalues have a descending order and represent the variance of corresponding principle components, while the eigenvectors stand for the projection directions that map the data from the original space to the principal space. By projecting the data to the directions of ***T***, we can acquire ***P***. The relationship between ***R, T,*** and ***P*** can be modelled as follows:

$$\boldsymbol{R} = \boldsymbol{T}\boldsymbol{P}' + \boldsymbol{E}, \tag{2}$$

where ***E*** is the reconstruction error when principal components with lower variance are ignored.

By comparing Eq. 1 and Eq. 2, one can easily find that the principal components ***P*** correspond to concentration profile ***C***, the loading vectors ***T*** correspond to vibrational spectrum ***S***, and the reconstruction error ***E*** explains the additive noise ***N***. The fact is that, however, consider that PCA first looks for the direction along which the projected data has the largest variance, then find directions that orthogonal to the previous ones. Therefore, we could end up with loading vectors that have negative values since there is no constraint of non-negativity. In the real situation, however, the spectra would always be positive.

That being said, the key problem of PCA is that it only decorrelates the data, which is not strong enough to retrieve the hidden features (see Fig. 6 (b), (c)). Therefore, to recover spectrum ***S*** and concentration ***C***, more severe constraints than irrelevance should be imposed.

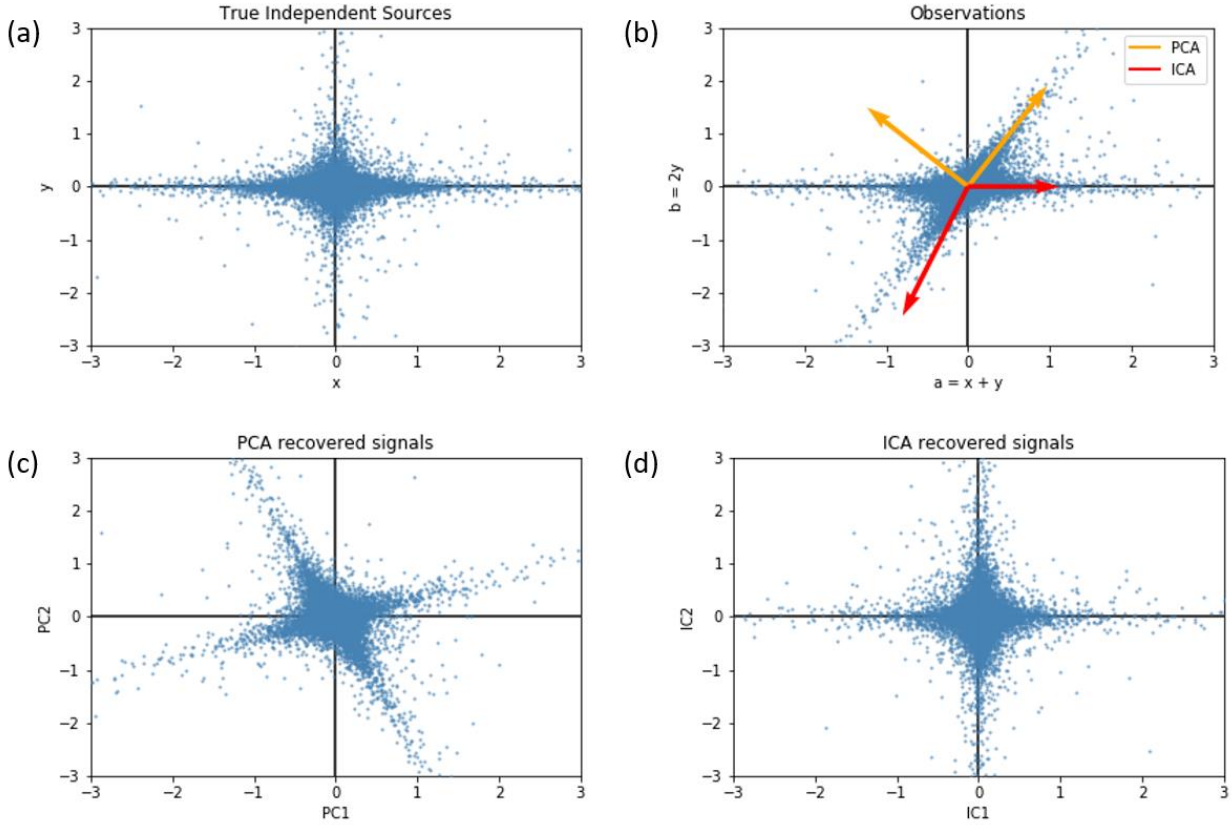


Fig. 6. (a) Two independent variables with student T distribution. (b) observations of mixed signals and the projection of PCA and ICA. (c) Distribution on PC space. (d) Distribution on IC space[6].

3.2.3. Independent component analysis (ICA)

As mentioned above, usually PCA is incapable of obtaining physically interpretable results, and a stronger statistical constraint than non-correlation is needed. ICA serves as an efficient algorithm for solving the “blind separation” problem, in which we need to unmix the linearly mixed source signals from observations.

The basic assumption of ICA is that the source signals are not only uncorrelated but also independent of each other. According to the central limit theorem, the properly normalized sum of $\boldsymbol{N}$ independent variables tends to have a normal distribution when $\boldsymbol{N}$ is large enough. So loosely speaking, less gaussian, more independent. Therefore, we could retrieve the independent variable by optimizing the non-Gaussianity of the variables.

ICA adopts the same linear-mixing model as mentioned above:

$$\boldsymbol{R} = \boldsymbol{A}\boldsymbol{s}, \tag{3}$$

where $\boldsymbol{s}$ is a matrix of original signals, $\boldsymbol{A}$ is the mixing matrix, and $\boldsymbol{R}$ is the mixed data from observations. The goal of ICA is to acquire both $\boldsymbol{A}$ and $\boldsymbol{s}$ from $\boldsymbol{R}$ by assuming that original signals are independently distributed.

First, $\boldsymbol{R}$ needs to be centered and whitened to have a unit variance to facilitate optimization.

$$\boldsymbol{z} = \boldsymbol{GR}. \tag{4}$$

$\boldsymbol{G}$ is a linear transformation that can be given by

$$\boldsymbol{G} = \boldsymbol{D}^{-1/2}\boldsymbol{V}^{T}, \tag{5}$$

where $\boldsymbol{V}$ and $\boldsymbol{D}$ are the eigenvectors and eigenvalues of the covariance matrix of $\boldsymbol{R}$. In fact, the whitening procedure is achieved by PCA.

After the preprocessing, a typical algorithm called fastICA measures the non-gaussianity by negentropy and used the fixed-point algorithm to achieve fast convergence[7] . The steps are specified as follows:

i. Initialize a random vector $\boldsymbol{w}$ of unit norm.
ii. Update $\boldsymbol{w} = \boldsymbol{E}\{\boldsymbol{z}\boldsymbol{g}(\boldsymbol{w}^{T}\boldsymbol{z})\} - \boldsymbol{E}\{\boldsymbol{g}'(\boldsymbol{w}^{T}\boldsymbol{z})\}\boldsymbol{w}$.
iii. Normalize $\boldsymbol{w}$.
iv. Repeat step ii and iii. until converged. $\boldsymbol{w}$ gives the IC spectrum.

Where function $\boldsymbol{g}(\boldsymbol{x})$ measures the negentropy of variable $\boldsymbol{x}$ and the higher the negentropy, the more non-Gaussian the variable is. A common choice of $\boldsymbol{g}(\boldsymbol{x})$ can be the hyperbolic tangent function. The projection directions of ICA are shown in Fig. 6(b), and we can find that ICA tends to project the original data to the directions that maximize the non-Gaussianity of the new variables. Fig. 6(c) shows the whitened data and Fig. 6(d) shows the projection of the original data points in the IC space.

3.2.4. Vertex component analysis (VCA)

In VCA, every pixel of the mixed data $\boldsymbol{R}$ is viewed as a point located at the $\boldsymbol{L}$-dimensional Euclidean space, and $\boldsymbol{L}$ indicates the number of channels. In this way, all the pixels form a point cloud at that space. Ideally, the cloud should locate within a simplex whose vertices are defined by pixels of the pure components, since values of other pixels are the linear combination of the purest ones. However, due to artifacts and noise, the cloud instead gets more dispersed (Fig. 7(b)).

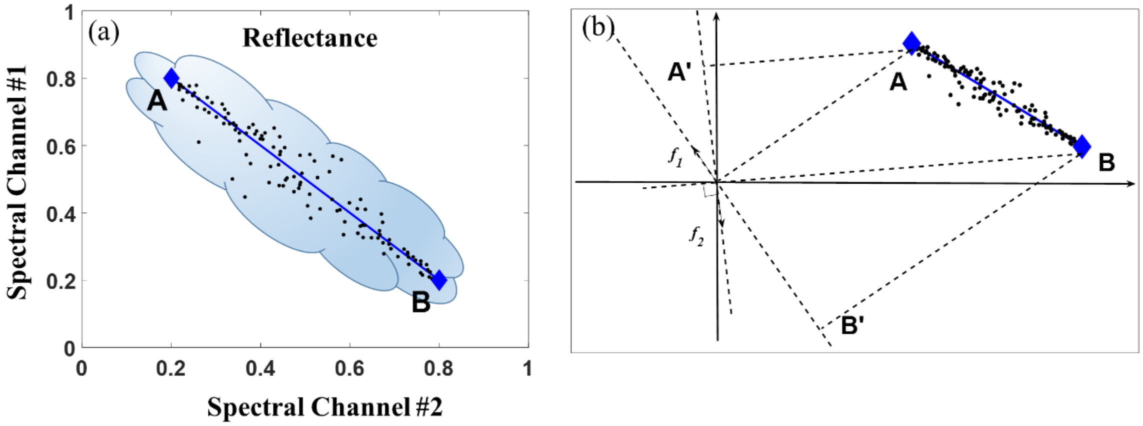


Fig. 7. Scheme of VCA (a) Two-dimensional scatterplot of two components (noted with A and B). (b) Illustration of VCA algorithm[8].

The model of VCA is almost the same as that described by Eq. 1:

$$\boldsymbol{R} = \boldsymbol{S}\gamma\boldsymbol{C}' + \boldsymbol{E}, \tag{6}$$

where γ is a scalar used to model the illumination variability due to surface topography. Fig. 7(a) shows the distribution of data cloud in two-dimensional space. The points should have been located on the line section whose two vertices are defined by the two pure components A and B, but instead, the points stray away from the line like a cloud due to noise.

Steps of VCA are shown in Fig. 7(b). First, the data cloud is projected back into a simplex by PCA or SVD (singular value decomposition) so that the effect of illumination and noise are partly depressed, and then the data is iteratively projected into the space orthogonal to the components already determined until all the components are exhausted. The first direction is randomly selected. The spectrum of the purest components corresponds to the extreme of every projection.

Therefore, VCA finds the purest points (pixels) and picks up the spectral data from these points directly. Compared with ICA, VCA removes ambiguities in the sign and the intensity of the spectrum. However, VCA subjects to “the pure pixel assumption,” which means if there are no pure pixels in one image stack, VCA cannot find the spectrum for pure components, but instead the mixed spectrum of the supposedly pure pixels by VCA.

4. Evaluation of spectral unmixing methods

In this chapter, I compared the unmixing results of the four algorithms aforementioned, i.e., PCA, ICA, VCA, and MCR. Furthermore, the performance of these methods was evaluated under noisy situation, where Gaussian noise with different power were added into original datasets.

The same dataset was applied to evaluate the four methods. The data were taken from a Hela cell, as shown in Fig. 8(a). There are mainly two components of interest in the cells: lipids and protein. For the reference, I first extracted spectra for both components manually and then decompose the data with the extracted spectra to get the concentration distribution.

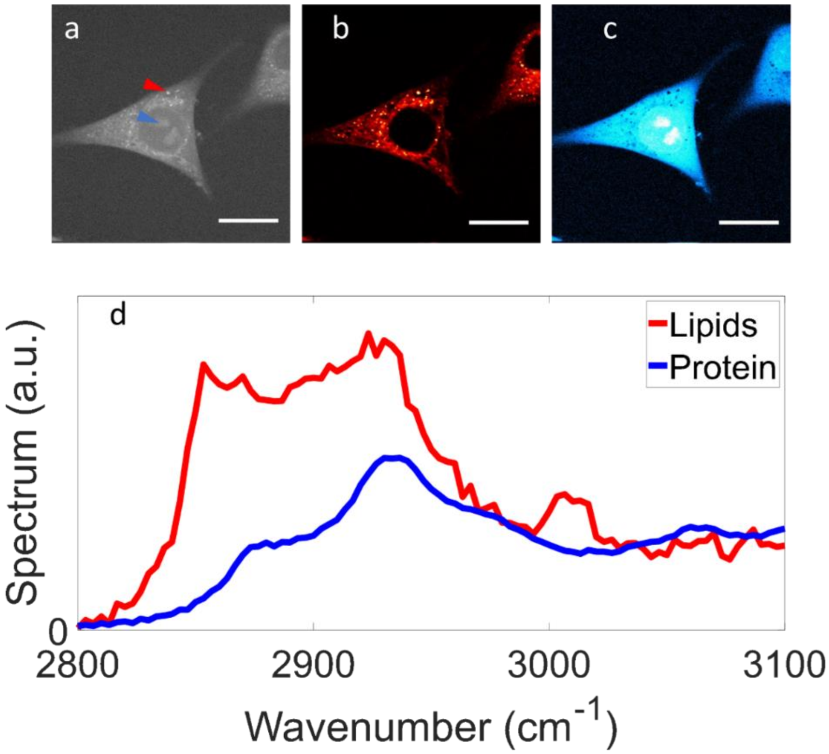


Fig. 8. Results of manual decomposition. (a) SRS image@2935cm^{-1}, red arrow, and blue arrow indicate the extraction point of lipids and protein (b) Distribution of lipids. (c)Distribution of protein. (d) Manually extracted spectra. Scale bar: 20 µm

We can see from the figure above that there is an obvious difference between lipids spectrum and protein spectrum. By using the extracted spectra, we can further decompose the raw image into lipids concentration map and protein concentration map. We can also know that lipids mainly distribute in the cytoplasm and has a relatively high intensity and more concentrated distribution. While protein extensively exists in the cell and has a high concentration in nucleoli.

4.1. PCA and ICA

Then PCA was applied to the same SRS dataset. The loading vectors of the first two principal components (PC) are shown in the Fig. 9. We can see that the loading vector of PC1 seems to be the protein, which accounts for the largest variance of the raw data, while PC2 is somehow similar to lipids. However, since only orthogonality was used when PCA was trying to find PC2, there are negative values in the spectrum of PC2, which is not physically interpretable.

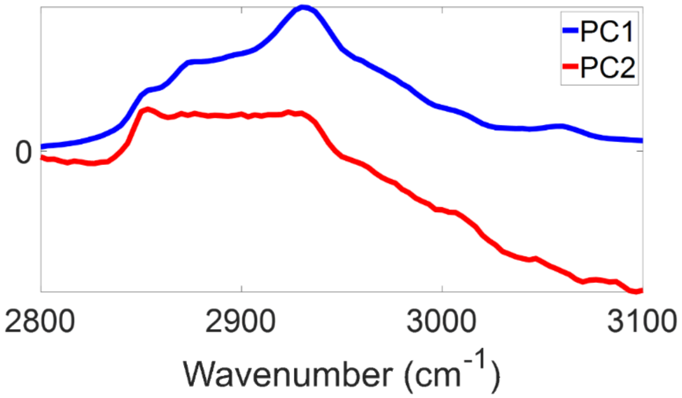


Fig. 9. Loading vectors (spectra) of the first two principal components.

PCA only decorrelates the mixed data, and it is not enough to unmix it to get spectrum and concentration profile. ICA, however, assume that variables, in our case, the distribution of the intensity of the lipids and protein, are statistically independent with each other. Consider that ICA can find the hidden variables with the largest impendence. If the assumption holds, we can hopefully recover the lipids and protein. The spectra of independent components (IC) are shown in Fig. 10.

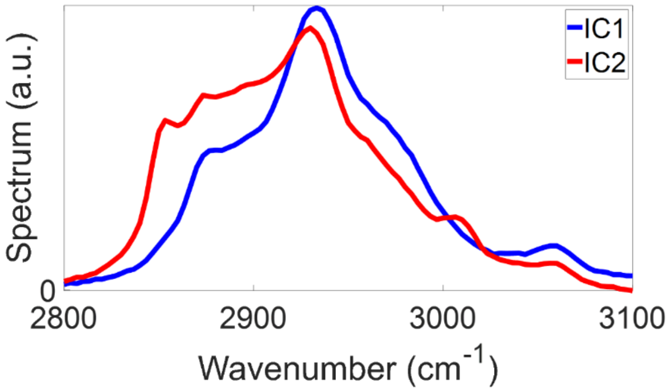


Fig. 10. Spectra of the independent components.

By comparing Fig. 9 and Fig. 10, we can find that ICA successfully removes the negative part in the spectrum of PC2. IC1 seems to be protein and IC2 looks like lipids. It is worthy to note that, however, ICA does not care about the order and the magnitude of independent components. The ICs could come out at random order and magnitude. Fortunately, the order and intense of ICs are not our concern.

4.2. VCA

As we have mentioned before, instead of exploiting the statistical information of the dataset, VCA tries to find the "purest pixels" by iterative projections. We could infer that VCA would have a similar manner of selection as our human because we also try to spot the "purest pixels." The spectra extracted by VCA are shown in the Fig. 11. We can see that the VC spectra do look like the manually extracted ones (Fig. 8): VC1 resembles lipids, and VC2 resembles protein.

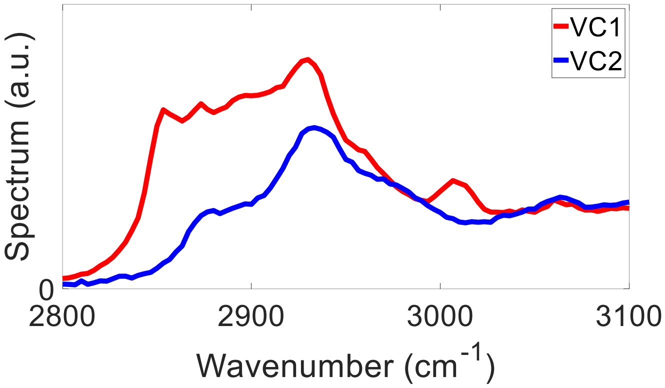


Fig. 11. Spectra of vertex components (VC)

It is worthy of mentioning that, to get more smooth spectra, instead of selecting only one purest pixel for each type of component, we chose multiple pixels for every class and then averaged the pixels to get the spectrum. Fig. 12 indicates the locations of pixels being selected for VC1 and VC2.

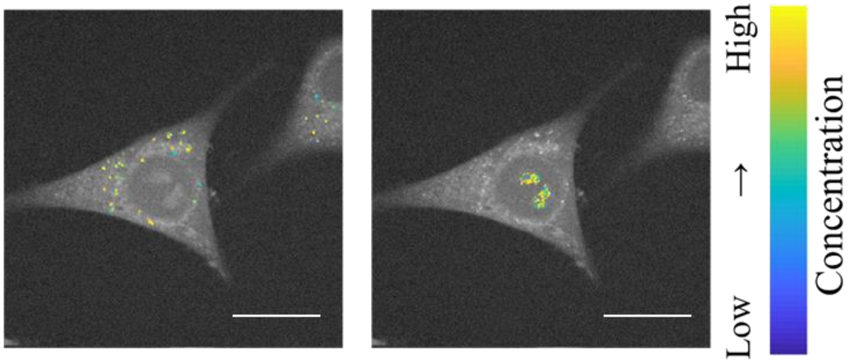


Fig. 12. the first 200 purest pixels selected by VCA, left: lipids; right: protein. Scalebar: 20μm

It seems that VCA could help us automatically find the spectra and then decompose the SRS dataset similar to the manual way. However, if there were no pure pixels in the image, VCA, as well as the manual selection, would not give us the pure spectrum. It is because no matter for VCA or manual selection, they can only extract the spectra from the existing pixels.

4.3. MCR

As mentioned before, MCR tries to impose extra physical constraints on the solution so that it could produce physically interpretable results. In our case, non-negativity was applied, and we tried to find the positive solutions that minimize the reconstruction error. Alternating least squares (ALS) was used to find positive solutions[9]. ALS find the $\boldsymbol{S}$ and $\boldsymbol{C}$ that minimize the reconstruction error in the least square sense iteratively and set the negative value to 0 in each iteration. Since only non-negativity was used, MCR evolves into another algorithm known as non-negative matrix factorization (NMF). It is worthy of mentioning that, however, since there could be multiple local minimums of the objective function, NMF could give us different results every time. Fig. 13 shows the spectra given by NMF. The red one resembles the lipids spectra, and the blue one resembles the protein one.

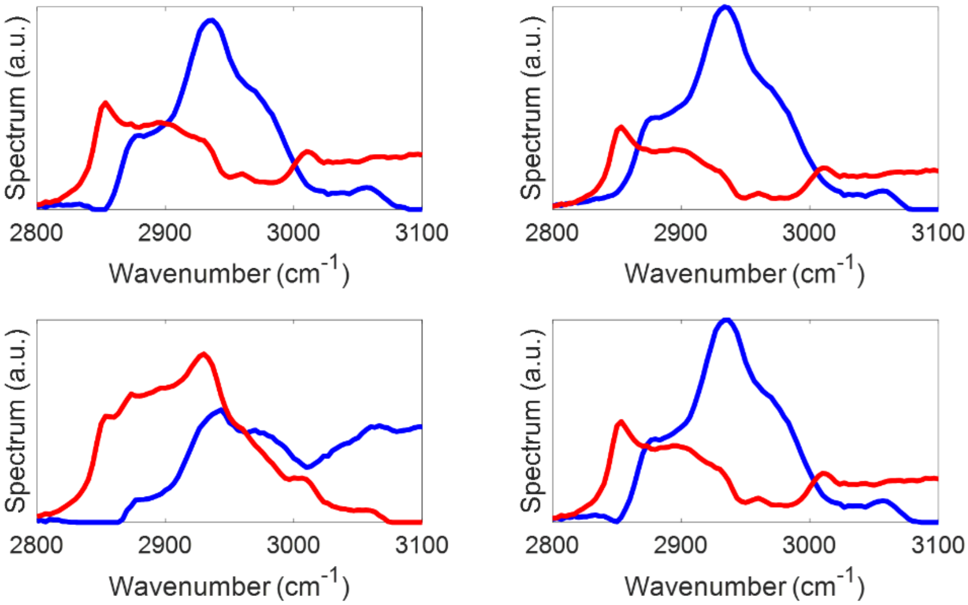


Fig. 13. Spectra extracted by NMF for four times.

We can see from the figure above that every time NMF gives different results. Regarding that every time NMF generates a random initial value, by fixing the initial value, we could reduce the randomness to a certain extent. However, compared to the manually extracted spectra, there are still relatively large differences. It seems that the constraint of non-negativity is also too weak to give us a satisfying result.

4.4. Comparison under different signal-to-noise ratio (SNR)

We have applied PCA, ICA, VCA, and MCR to the same dataset to evaluate the performance. It turned out that there is a certain consistency in the results. Another important aspect, however, is how they perform under low SNR. It is because sometimes, due to the effect of artifacts and systematic errors, we cannot get high-quality SRS images. Therefore, it is necessary to examine the performance of the algorithms under different SNR. Since the PCA spectrum is not physically interpretable, only ICA, VCA and MCR are compared under noise.

For a specified SNR, the Gaussian noise with corresponding power $\boldsymbol{E_n}$ was added to every pixel. $\boldsymbol{E_n}$ was determined by the following equations.

$$\boldsymbol{E_s} = \frac{\sum_{\boldsymbol{N}} \boldsymbol{R}^{\wedge\boldsymbol{2}}}{\boldsymbol{N}} \tag{7}$$

$$\boldsymbol{E_n} = \frac{\boldsymbol{E_s}}{\boldsymbol{SNR}} \tag{8}$$

$\boldsymbol{E_s}$: the power of the signal
$\boldsymbol{R}$: SRS datacube
$\boldsymbol{N}$: the number of pixels in $\boldsymbol{R}$

after adding Gaussian noise with different power, images at 2935cm^{-1} are shown in the Fig. 14.

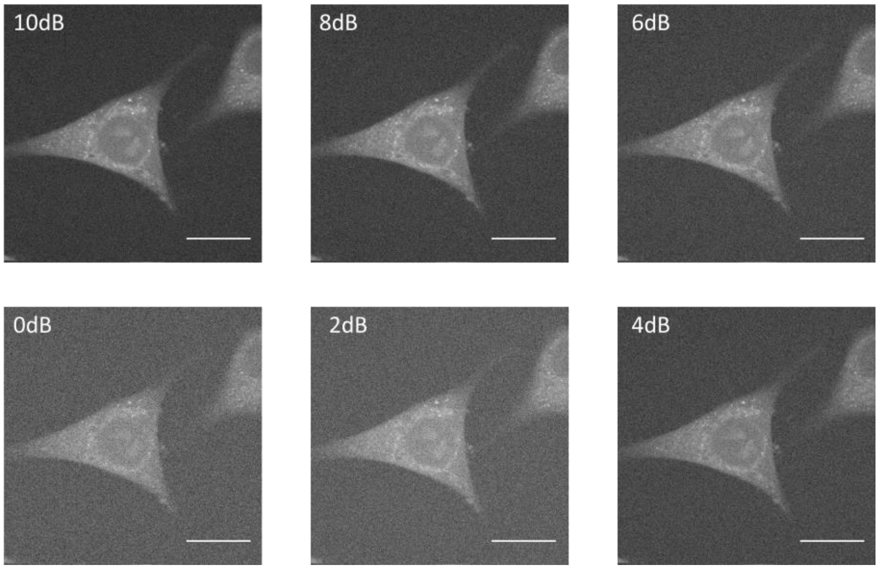


Fig. 14. images @ 2935cm^{-1} with different SNR level. Scalebar:20µm

Then I applied ICA, VCA, and MCR to the noisy data. To measure the performance of the algorithms, the distance between the spectra extracted under noise and the spectra unmixed without adding noise was calculated. Mean relative absolute error (MRAE) was selected to measure the distance. Where

$$\boldsymbol{MRAE} = \boldsymbol{MEAN}\left(\frac{|\boldsymbol{S_{noise}} - \boldsymbol{S_{ori}}|}{\boldsymbol{S_{ori}}}\right) \tag{9}$$

$\boldsymbol{S_{noise}}$: spectra extracted after adding noise

$\boldsymbol{S_{ori}}$: original spectra without adding noise

To ensure generality, around ten samples were tested. There exists consistency in the results for the ten samples, and two of them are shown in the Fig. 15. It is worthy of mentioning that I used the VCA spectra as the initial input for MCR to stabilize the solutions.

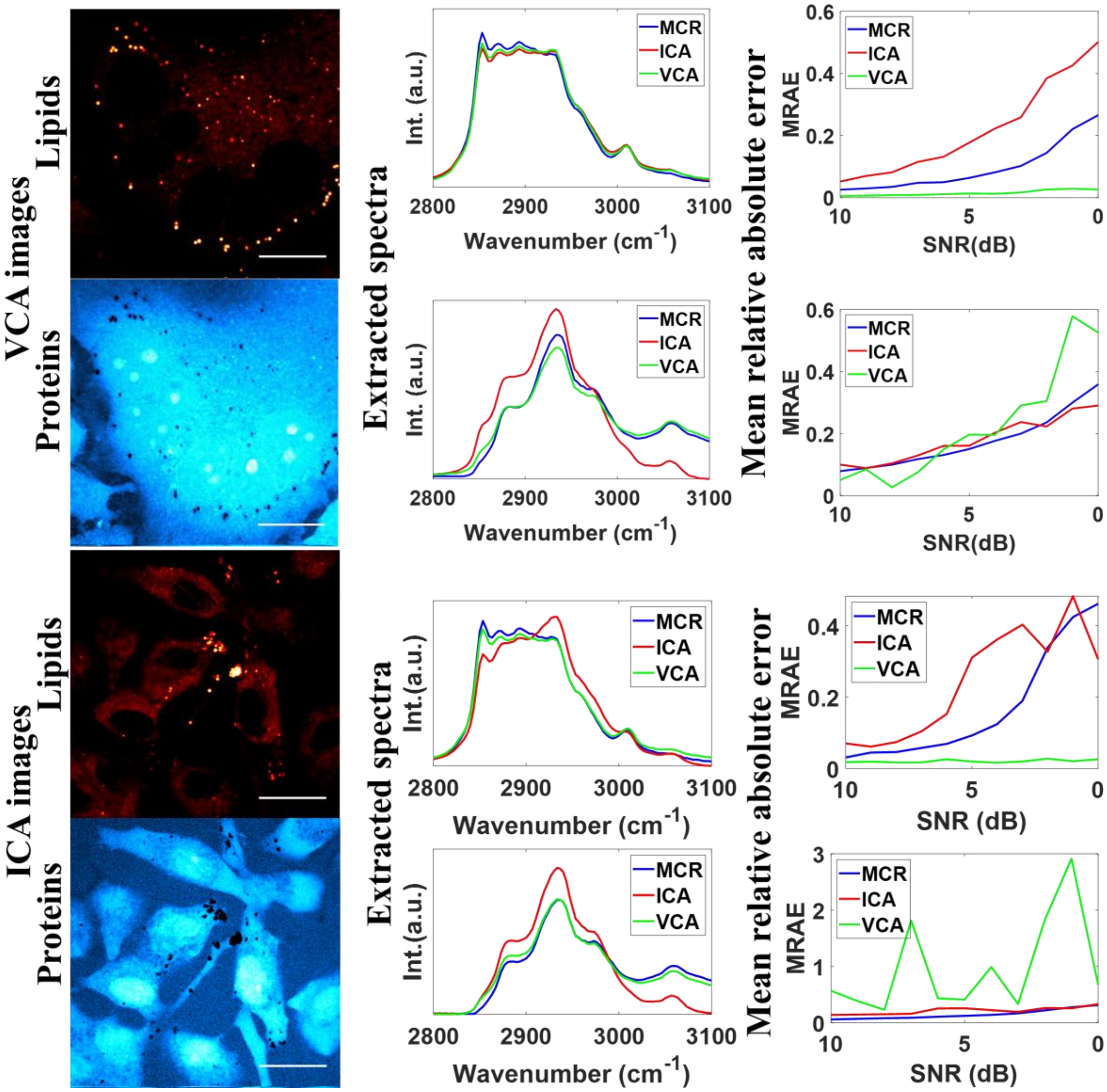


Fig. 15. Performance of MCR, ICA, and VCA under different SNR. Scalebar:20μm

We can find an interesting comparison form the figure above VCA performed much better for lipids under noise while ICA and MCR have a more stable performance for protein. Besides, it is also worthy to note that for the protein spectrum, ICA has lower tail than VCA and MCR. It is likely that the water spectrum that has a higher tail that influences the MCR and VCA spectra. ICA, however, is free from the influence of water since it seeks the independent signals.

A possible reason behind could be that VCA tries to pick up protrusive pixels. Consider that lipids pixels have a high signal intensity and thus should be "standing out" from the rest pixels and remain a relatively high SNR even though the noise level increases. Therefore, VCA could still easily find the lipids pixels and give barely unaffected spectra when SNR decreases. For less protrusive protein, however, VCA suffers a lot when noise is growing larger. ICA and MCR, however, try to have a global optimization thus are not so good at utilizing local information, but it also enables a relatively stable performance compared to VCA.

4.5. Summary

In conclusion, I comprehensively investigated four algorithms—PCA, ICA, VCA, MCR for unmixing hyperspectral SRS datacube. There is two key information that we want to recover from the mixed data $\boldsymbol{R}$, i.e., the vibrational spectra $\boldsymbol{S}$ and concentration map $\boldsymbol{C}$. They satisfy a liner relationship $\boldsymbol{R}=\boldsymbol{S}\boldsymbol{C}'$ (with noise ignored).

PCA tries to decorrelate data by finding the direction that accounts for the maximum variance, but the noncorrelation is not enough to give us physically interpretable results. ICA further pushes the boundary to independence and tries to recover the variables that are independent of each other. It gives us reasonable results while still could be questioned since we do not know if the concentration of different components is independent in a statistical sense.

Instead of exploiting statistical information like PCA and ICA, VCA takes advantage of geographical information of the data. It seeks the purest pixels by finding the vertices of the cloud comprised of data points. However, if there were no pure pixels, VCA would not be able to separate the pure spectrum from the mixed data. That is why we can find a possible influence of water on the protein spectrum extracted by VCA in Fig. 15, it is because protein and water could always be mixed.

Finally, MCR seeks the solutions that minimize the reconstruction errors under the non-negative constraint. Sometimes it could also give us reasonable results while it suffers from the local minimum and appears to have different results every time Fig. 13. By fixing the initial value, we could decrease the randomness to a certain extent. Furthermore, ICA, VCA, and MCR were compared under the noisy situation. It turned out VCA can recover lipids even under low SNR while ICA and MCR tend to have more stable performance than VCA. A plausible explanation is that VCA has local selectivity while ICA and MCR do global optimization. Therefore, pixels with high intensity like lipids can be easily found by VCA than protein.

Since the manual extraction is time-consuming and subjective, finding automatic unmixing algorithms would be highly desirable. However, as we discussed, the algorithms have both advantages and disadvantages. A possible combination could be that VCA for lipids and ICA for protein because VCA has a high selectivity for lipids pixels while ICA can separate protein form protein-water mixtures.

All the algorithms mentioned above could be categorized to unsupervised learning to a certain extent. It is because the algorithms need not training data to learn features first, but unmix the data by the features predetermined: PCA seeks the largest variance, ICA finds independence, VCA recovers purest pixels while MCR retrieves non-negative solutions. However, there could be hidden features that remain undescribed. It could be hard to define them precisely. Therefore, to learn the features, CNN architecture will be introduced in the following chapter. CNN tries to learn the mapping between input and output in a nonlinear manner. It tries to do convolution both along spectral and spatial dimensions, which could learn the interactive features between the two dimensions.

5. Classification and regression with convolutional neural network (CNN)

In this chapter, I mainly discussed the applications of CNN in analyzing SRS images. There are two main applications here: segmentation of cell structures such as cytoplasm, nucleus, and nucleolus. Segmentation could help us quickly identify cell structures. A traditional way to do segmentation is by thresholding, which uses the difference of brightness to binarize the images. In one of our experiments, we use the Gaussian filter and Otsu method[10] to make masks for blood cells (See appendix). We also combined the spectral unmixing methods and thresholding algorithms to improve the performance (See appendix). From the two experiments, we can see that with the thresholding algorithm, we achieved a reasonable segmentation of cells and background, and then build cell libraries at the single-cell level.

However, when the cell structure is complexed, and there are various components in the cell, the illumination tends to be uneven, and brightness-based thresholding algorithm could suffer. By learning multiple hidden features, CNN could give a better performance.

The second application is to let CNN simulate the unmixing algorithms. VCA results were used to train a CNN model, and we found that a pre-trained model can enable a fast separation of lipids and protein given quite limited spectral channels.

The basic CNN model used for both applications is known as U-net, and we will have a look at it in the following section. In this chapter, all the CNN codes were programmed in Python using the Keras library with TensorFlow as background and ran on Nvidia GTX 1070 8GB.

5.1. U-net architecture

Convolutional neural network (CNN) has been used as the main technique in object recognition contests sine 2012[11]. It is a kind of supervised machine learning technique that learns the mapping between input and output in a nonlinear manner. After the learning is done, it can be used in predicting new data. It turned out that CNN with deep layers and smaller convolution kernels tend to perform better[12]. A typical deep CNN architecture used in cell segmentation is known as U-net[13]**Error! Reference source not found.**. The structure of U-net is shown in Fig. 16.

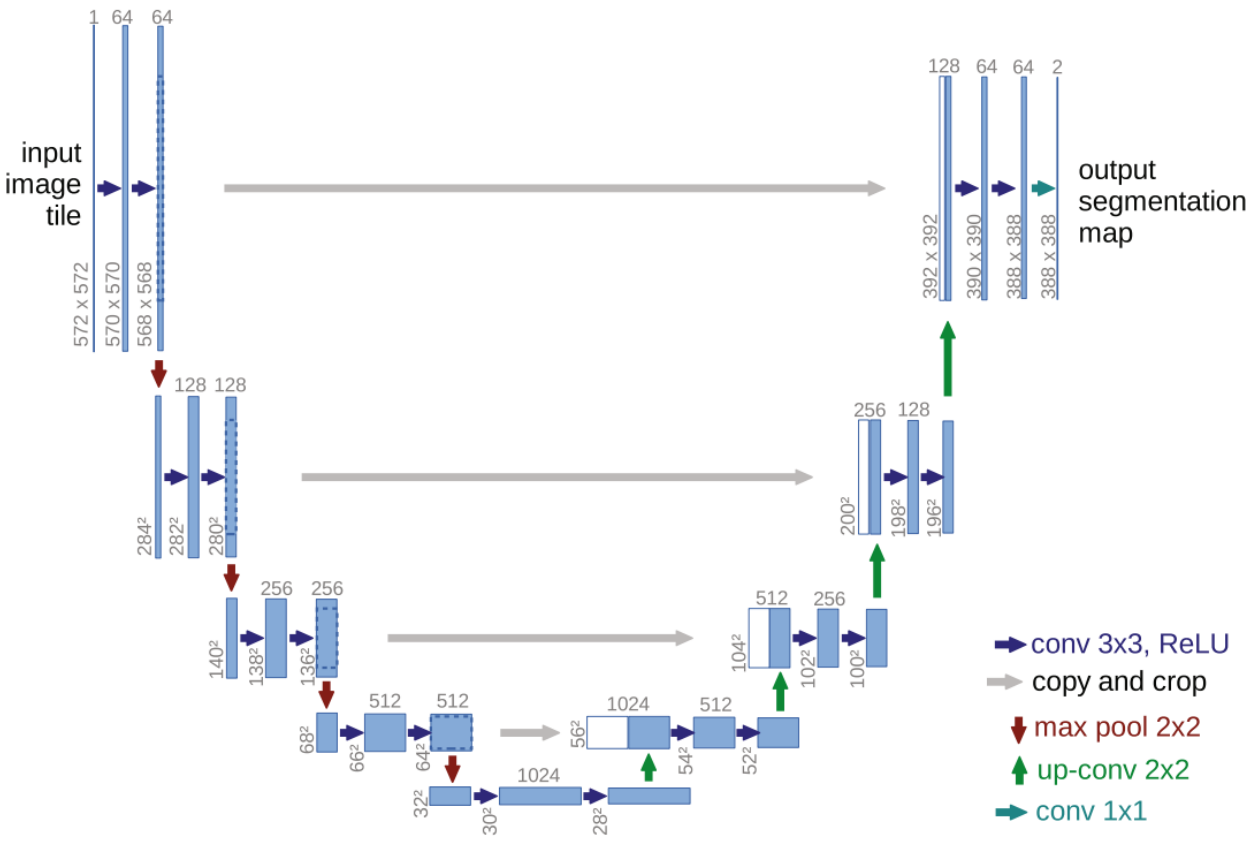


Fig. 16. U-net structure

U-net consists of the encoding part (the left branch) and the decoding part (the right). The shape of U-net is similar to the letter "U," and that is why it is called "U-net." There are two computational advantages of U-net: 1) there is no fully connected network (FCN) in the architecture. Therefore it avoids the high computational cost of FCN. 2)The size of tensors gets smaller flowing down through the encoding way and thus the size of parameters to be trained get smaller compared to the architectures without down-sampling. Besides, it is also worthy to note that the shallower layers are concatenated to the deeper layers, in this way the training is short-circuited, and the flow of gradient could be facilitated. The information from the shallower layers could also be reserved in deeper layers.

U-net has a relatively smaller requirement for the size of training data: in the U-net paper, only 30 images and corresponding masks[14] were used in the training process. Three of the training pairs (training images and labels) are shown in Fig. 17. The upper row consists of three original images, and the lower row is the corresponding masks.

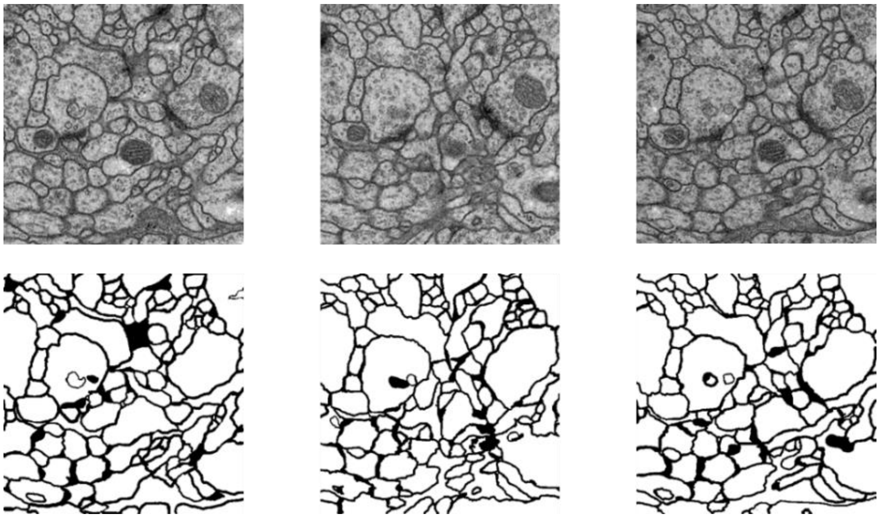


Fig. 17. Training datasets

I first repeated the demonstration: U-net was trained by the 30 training datasets with stochastic gradient descent algorithm and penalized with the cross-entropy function of the soft-max probabilities. Then the test data was input to the trained model for prediction. The results were shown in Fig. 18. It achieved a validation accuracy of 82.92%.

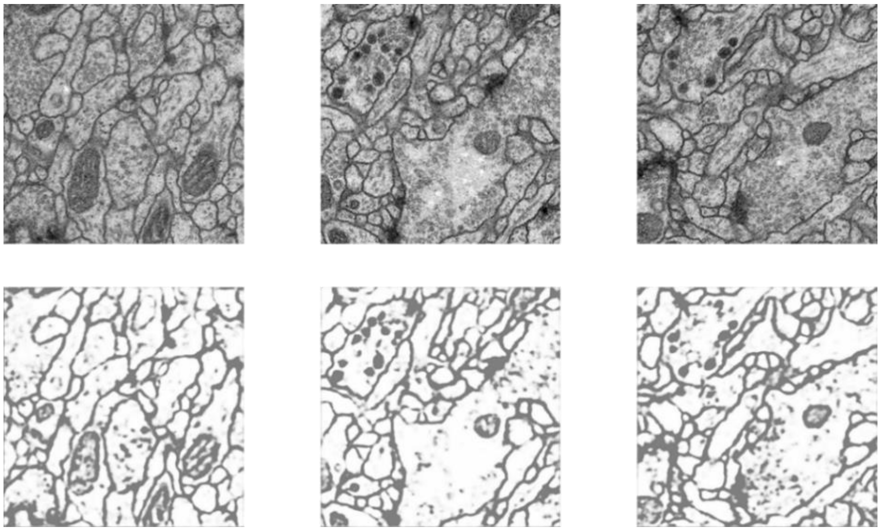


Fig. 18. Prediction of the demonstration data after ten epochs of training

The validation accuracy here means that we separated the 30 training datasets into two groups: 24 for training and another 6 for validation. The validation group did not participate in training but was used for prediction. Then the results were compared against the ground truth to calculate the accuracy, the so-called validation accuracy. The validation accuracy serves as a good indicator of whether the model works well on the new data.

Furthermore, the performance can be improved by the data augmentation, i.e., the distortion of the original images to increase the volume of the training data. The distortion involves the rotation, shift, flip, zoom, and shear of an image, as shown in Fig. 19.

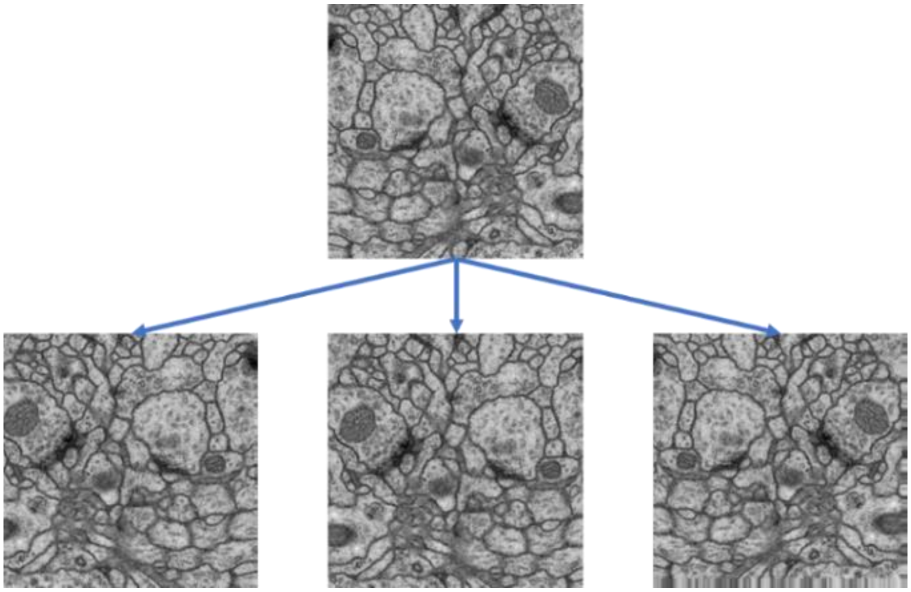


Fig. 19. Data augmentation

After the data augmentation, CNN got more robust against artifacts. After augmenting the training data by 30 times, i.e., 29 augmentation images were generated for each training image, and now there are 30*30=900 datasets available for training. The prediction results are shown in Fig. 20. The validation accuracy reached 91.70%.

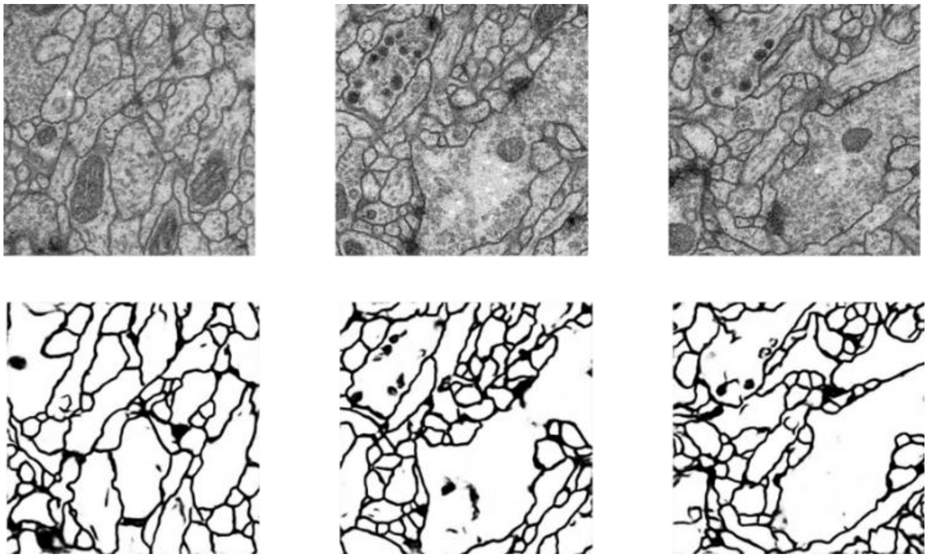


Fig. 20. Prediction results with the augmentation factor = 30, epochs = 10

5.2. Classification of cell structures

The U-net architecture could also be used in SRS dataset to help us locate cell structures. An intuitive solution is that one can select an SRS image with high SNR and then manually makes the corresponding mask and puts them together as a training dataset. Later after the model is trained, the trained model can be used in the further prediction.

First the I made masks of the nucleus. The reasons I chose nucleus are that 1) Manual masking is expensive. For our datasets (Hela cells), the nucleus is easier to be profiled than cell body itself. 2)Spectral unmixing algorithms can give us a rough shape of a cell body with protein concentration map; they cannot, however, locate the nucleus due to the extensive existence of water and protein in cells. Therefore, segmentation with CNN could help us get the location of the nucleus that is hard for spectral unmixing algorithms to acquire.

The SRS image at 2935 cm^{-1} was used as the train image due to its high signal SNR. From this image, we can find clear profiles of the nucleus.

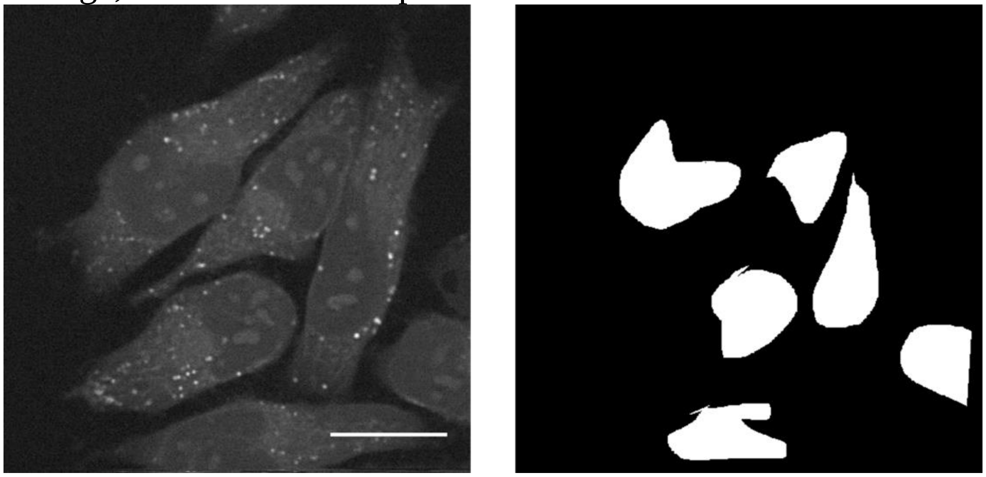


Fig. 21. SRS image @ 2935cm^{-1} and the corresponding nucleus mask. Labels were manually made by CellProfiler[15]. Scalebar: 20μm

In this way, 24 training data sets were made and were augmented by five times. Later the augmented datasets were used to feed the model. Even though the number of original datasets was quite small, the trained model achieved a relatively good performance by a validation accuracy of 93.67%. One of the prediction results is shown in Fig. 22.

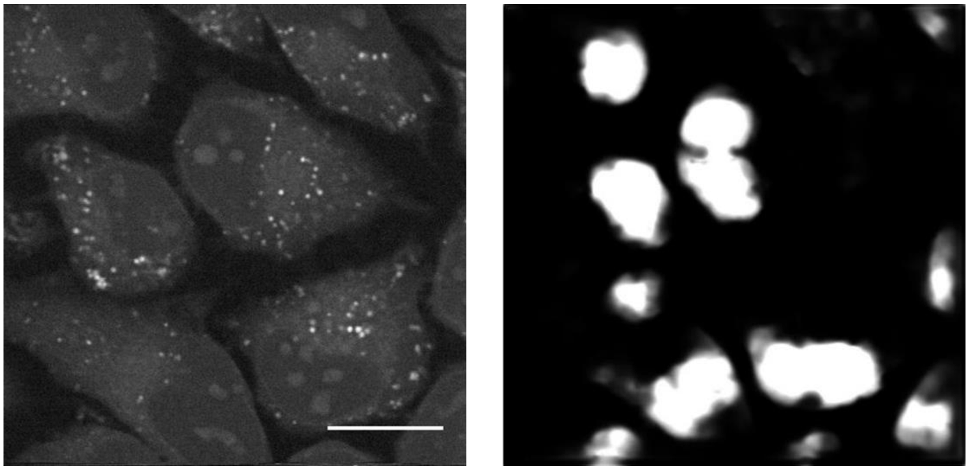


Fig. 22. Prediction results with 1-channel input. Scalebar: 20µm

Similar to the demonstration case, only spatial information was used here since we just used one SRS image. However, SRS is capable of imaging cells in a hyperspectral manner, so it could help improve the accuracy if we also incorporate the spectral information. A direct way would be selecting N channels from SRS data cube, thus would make the input training images as X-Y-N matrix, where X and Y correspond to the spatial dimensions and N corresponds to the spectral dimension. In this way, by taking convolution along the three dimensions, both spatial and spectral information can be exploited.

A tricky thing is which spectral channels should be selected. The original thought was to select the ones that maximize the sparsity; in this way, a good balance was supposed to be struck between computational resources and accuracy. However, due to a very limited GPU storage, maximum to 3 channels were allowed. Therefore, three channels were picked up properly to represent the spectral differences between the nucleus and cytoplasm. As shown in Fig. 23, the difference between the cytoplasm spectrum and nucleus spectrum looks like the spectrum of lipids. It is reasonable because the main difference between cytoplasm and nucleus is the existence of lipids. Finally, SRS images at 2853cm^{-1}, 2903cm^{-1}, and 3006cm^{-1} were selected as the input training dataset.

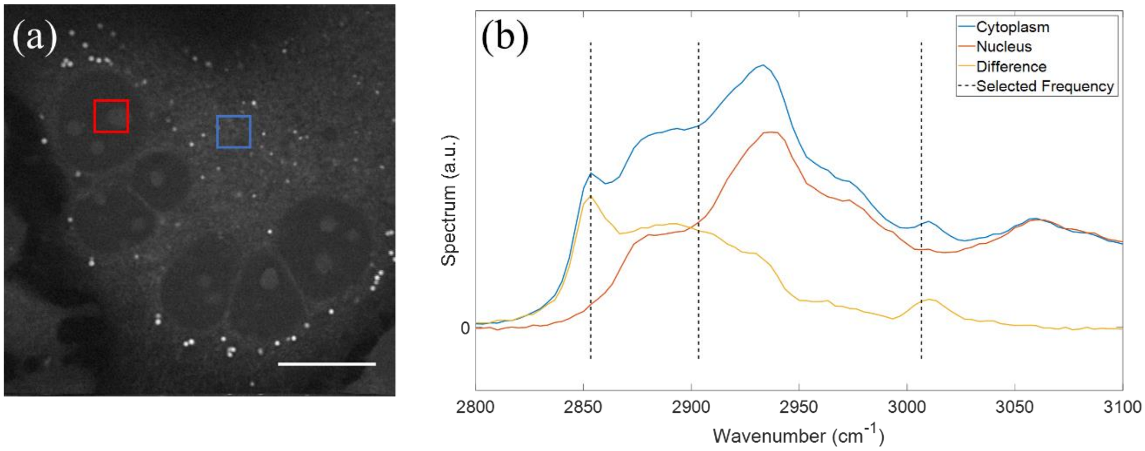


Fig. 23. (a) Selected nucleus part (red box) and cytoplasm part (blue box) for the spectral plotting. (b) Difference between cytoplasm and nucleus spectrum and the selected frequency. Scalebar: 20µm

Later, a dataset comprised of the three channels, and the corresponding mask of the nucleus was input as the training image.

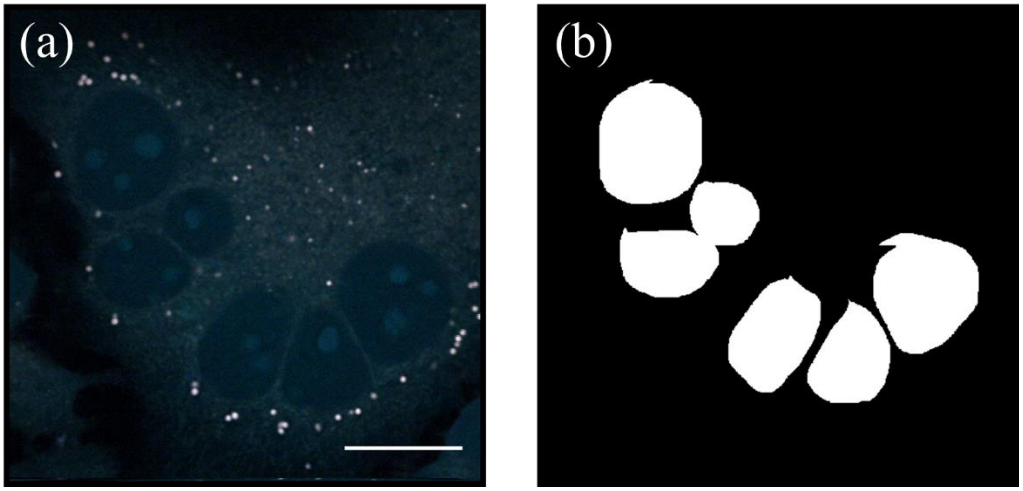


Fig. 24. (a) A training image comprised of SRS images @2853cm^{-1}, @2903cm^{-1}, and @3006cm^{-1} as red, green, and blue. (b)Corresponding mask of nuclei. Scalebar: 20µm

It turned out that the validation accuracy got improved (from 93.67% to 95.20%) from the 1-channel input. After training 24 datasets for 40 epochs, the prediction results are shown in Fig. 25.

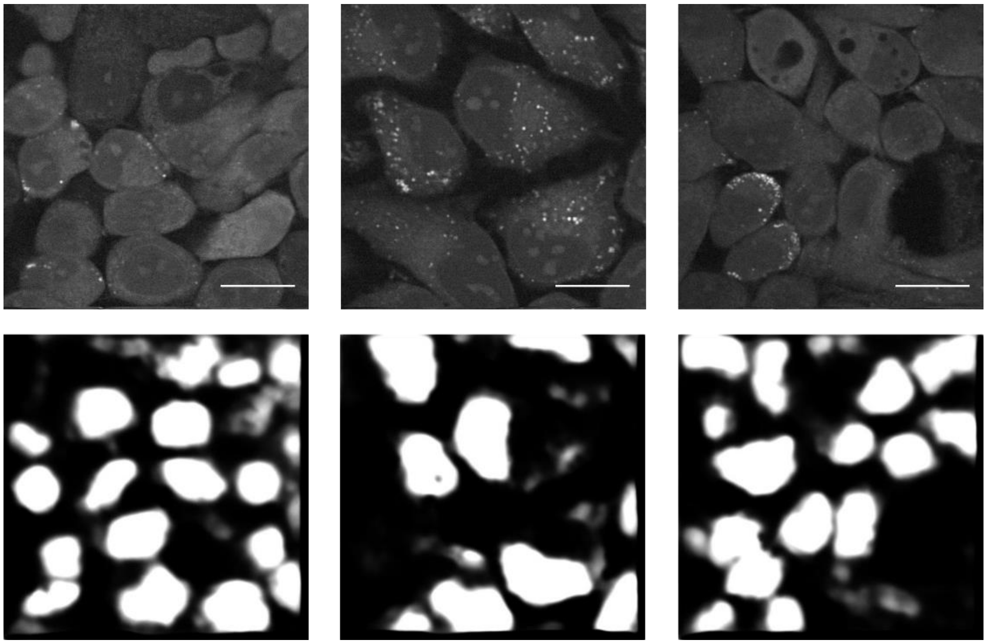


Fig. 25. Prediction results for 3-channel input. Scalebar: 20μm

Furthermore, if we also increase the dimension of label images, i.e., add more classes like cell body, nucleolus, we could expect that CNN would also do the multi-class classification. According to this notion, 4-class labels were made, including background(water), cytoplasm, nucleoplasm, and nucleolus.

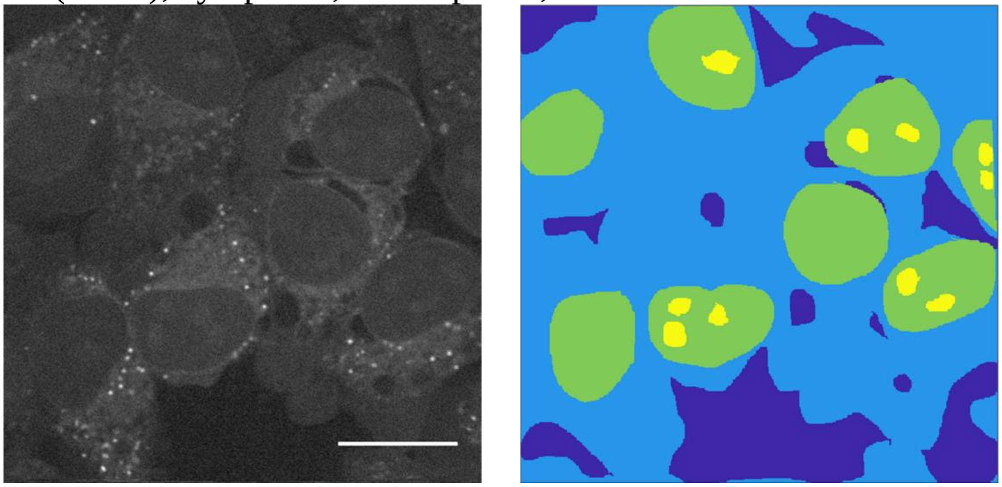


Fig. 26. The label image with four classes: water as deep blue, cytoplasm as cyan, nucleoplasm as green, nucleolus as yellow. The label image was manually made. Scalebar: 20μm

To make the SRS images more representative for all the four components, different channels were selected than before.

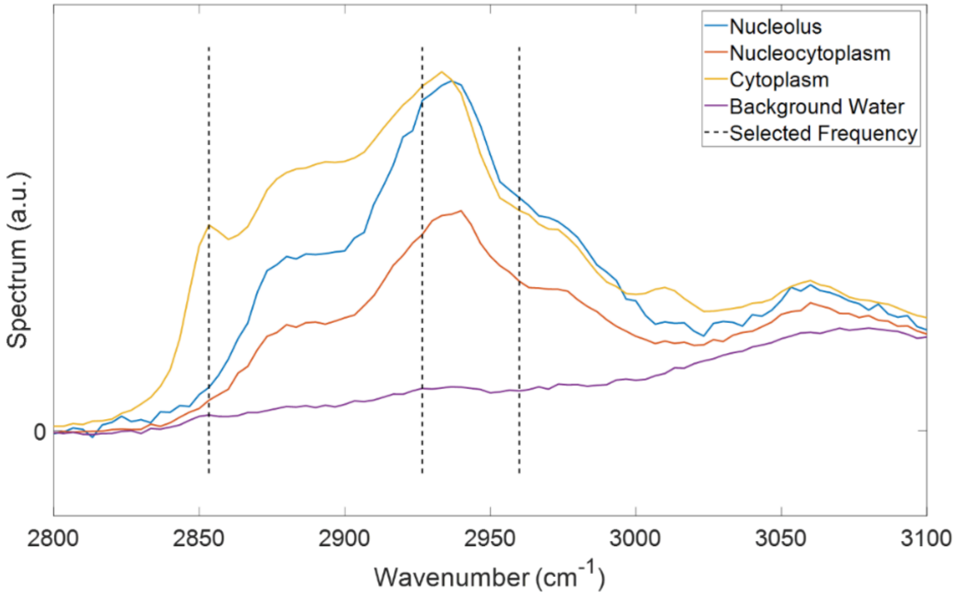


Fig. 27. Spectra for the four components and the selected frequency @2853cm^{-1}, @2926cm^{-1}and @2960cm^{-1}

Then the updated 3-dimensional SRS images were input as the training data, after being trained, the prediction of the new images are shown in Fig. 28.

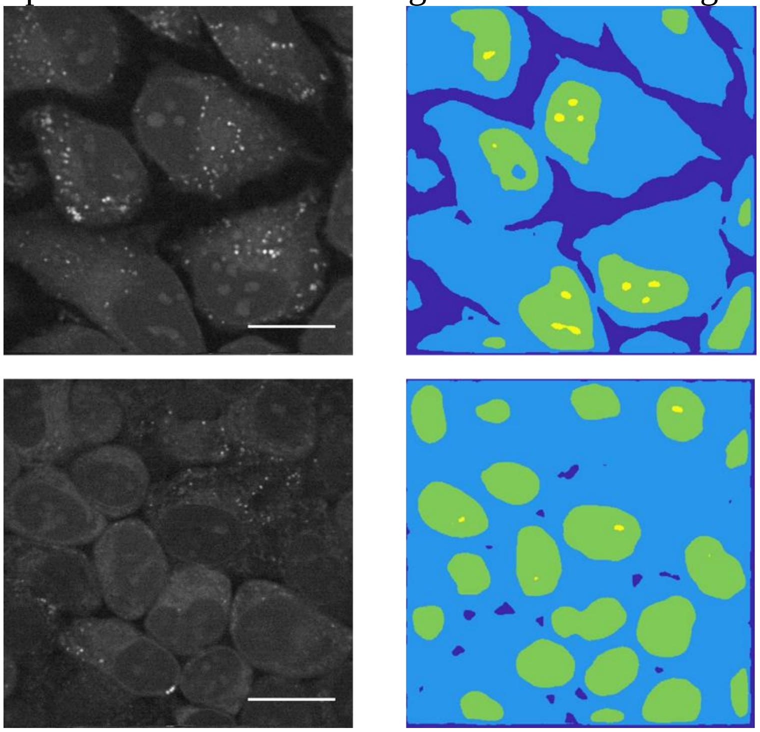


Fig. 28. Prediction results. Background water as deep blue, cytoplasm as cyan, nucleoplasm as green, nucleoli as yellow. Scalebar: 20µm

From the figure above we can find that CNN gave a quite accurate predication on the locations of background water, cytoplasm, and nucleoplasm. However, it seems somehow hard to find nucleolus. The reason could lie in the training images or the label images. Due to the difficulty to get more accurate nucleolus profiles manually, adjusting to the training dataset seems easier.

The concentration map acquired by spectral unmixing algorithms (Fig. 29), e.g., VCA or ICA, can also serve as the training images. The intuition behind it is that they give a clear separation of different components. However, the direct using of the component maps tends to have the nucleoplasm being classified as cytoplasm (Fig. 30). Perhaps due to the extensive existence of protein both in the cytoplasm and nucleus make CNN hard to discriminate the two structures.

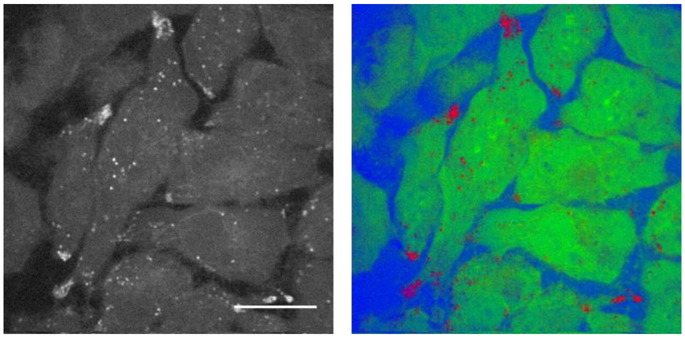


Fig. 29. (a)SRS image @ 2935cm^{-1} (b) Merged figure of VCA components: lipids as red, protein as green and water as blue. Scalebar: 20μm

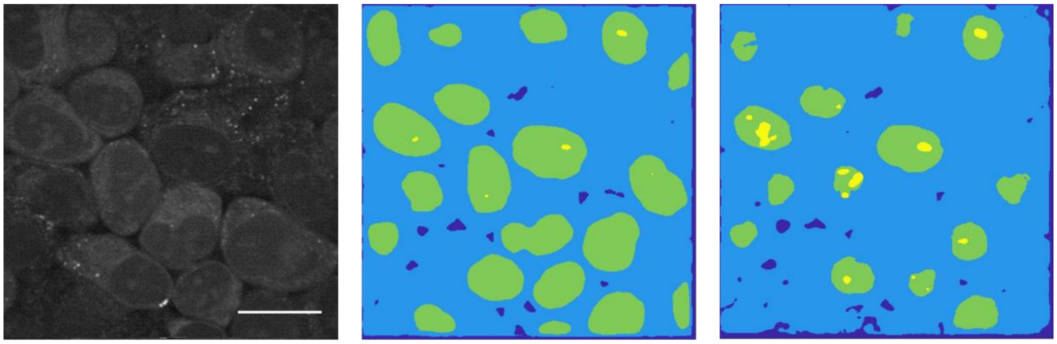


Fig. 30. (a)SRS raw image. (b) prediction results based on raw SRS images. (c) prediction results based on VCA components (lipids, protein, and water). Scalebar: 20μm

However, one interesting point of the VCA-based prediction is that, although the discrimination of nuclei seems not good, the locations of the nucleoli (Fig. 30 (c) yellow area) look more accurate than the raw-image-based prediction (Fig. 30 (b)). A possible explanation could be that the protein map of VCA highlights the nucleolus that has much higher protein concentration than the nucleoplasm.

Therefore, we could strike a balance between the nucleoplasm differentiation and nucleolus identification. A simple way is to concatenate VCA images and raw SRS images together as the input training dataset. In this way, both the component map and the original SRS data can be exploited. Since that it should be the protein concentration map that facilitates nucleolus localization, the protein concentration map, together with the first two SRS images @2853cm^{-1}, @2926cm^{-1} were combined as the training images (Fig. 31). The prediction results were shown in (Fig. 32). It turned out that this approach can get more accurate nucleolus profile without misclassifying nucleoplasm as cytoplasm as before.

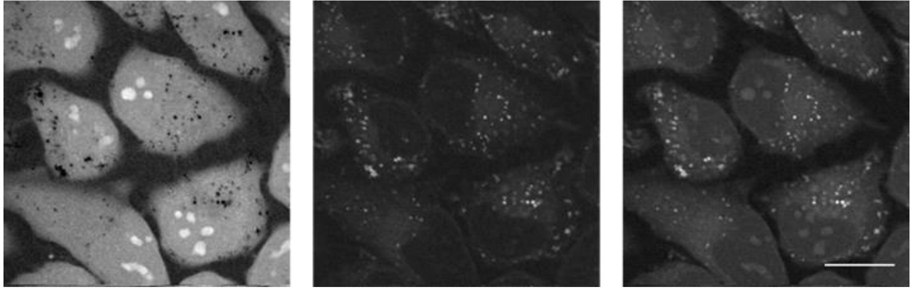


Fig. 31. mixed input from VCA protein concentration map (left), SRS image @2853cm^{-1} (middle), and (c) SRS image@2926cm^{-1} (right). Scalebar: 20µm

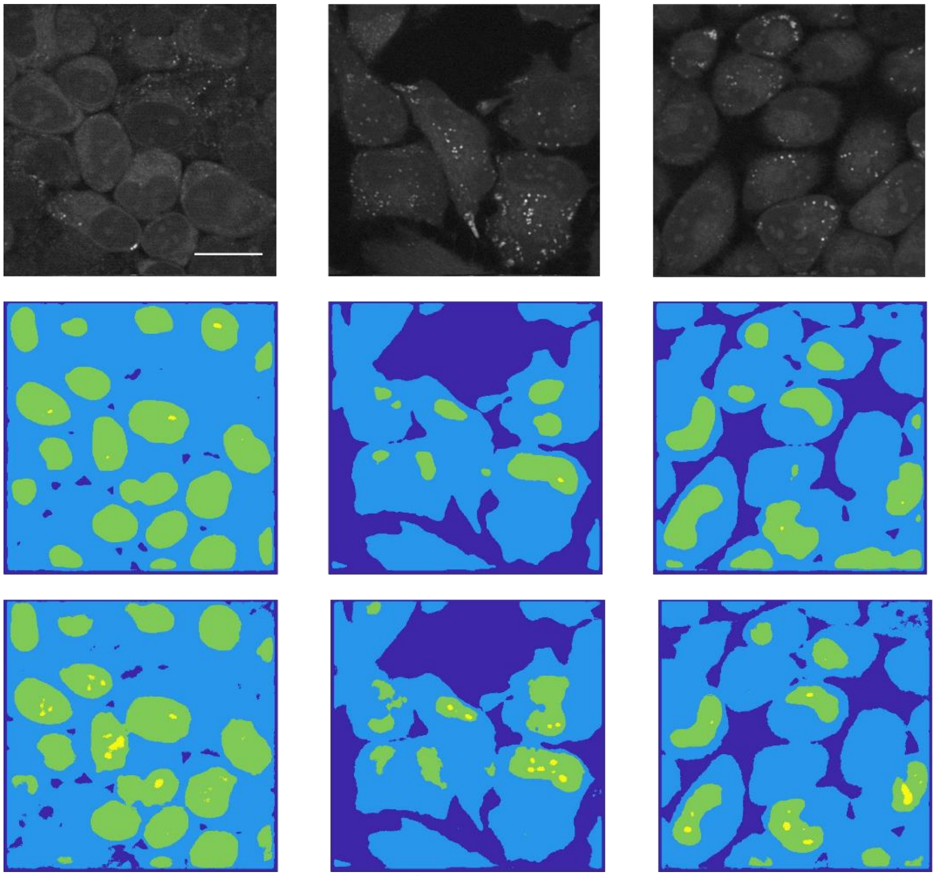


Fig. 32. 1st row: SRS image @ 2935cm^{-1}; 2nd row: raw-image-based prediction; 3rd row: VCA and raw images combined prediction. Scalebar: 20µm

Up to now, the CNN structure has achieved an amazing segmentation of cell structures, i.e., nucleolus, nucleus, cytoplasm, and background water. By flexibly choosing the training images from either raw SRS images or the concentration map attained from the spectral unmixing algorithms like VCA, the performance can be further improved.

However, asides from the training datasets, the quality, and quantity of label data are another essential part for the performance. Until now, the labeled images were acquired manually: we carefully outline the different structures. Although the manual labeling is not so accurate, CNN performed reasonably well. When the volume of training datasets increases, however, manual labeling would be quite time-consuming. Therefore, a reliable, automatic labeling approach would be of desire.

An alternative could be found biologically rather than algorithmically. Fluorescent imaging enables us to stain different cell structures with corresponding fluorescent materials. A staining chemical called NucBlue, which is commonly used to dye DNA, is used to mark the nucleus, given the abundant existence of DNA in the nucleus. If we first stain the nucleus and then do the SRS imaging and fluorescent imaging on the same field of view simultaneously, we could get the accurate location of the nucleus, as shown in Fig. 33. Later, since the nucleus in the fluorescent image has a relatively high SNR level, by a simple thresholding algorithm, one could get relatively accurate masks. Therefore, if we could make masks in this way, it would enable better profiling and thus better performance.

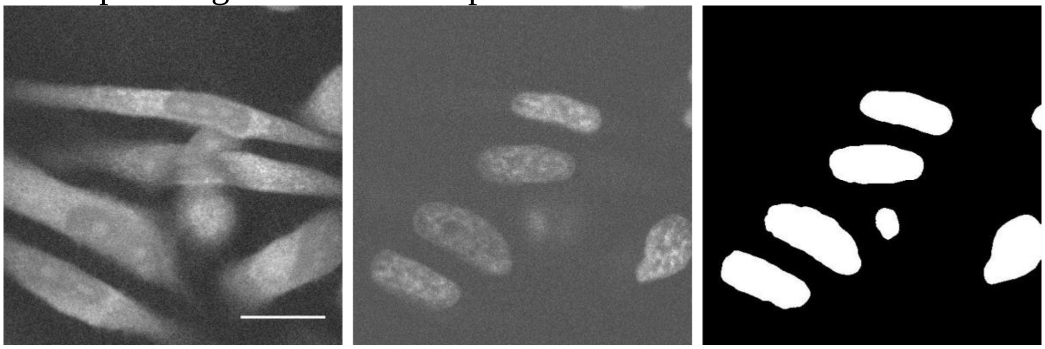


Fig. 33. (a) Hela cells, SRS image @2937cm$^{-1,}$ and the corresponding (b) fluorescent image of the stained nuclei. (c)masks of nuclei generated by thresholding algorithm. Scalebar: 20μm

We used the same thresholding algorithm, as mentioned in the appendix.

Later the SRS images and the corresponding masks were fed into the model. After training for 60 epochs, the prediction results are shown in the Fig. 34.

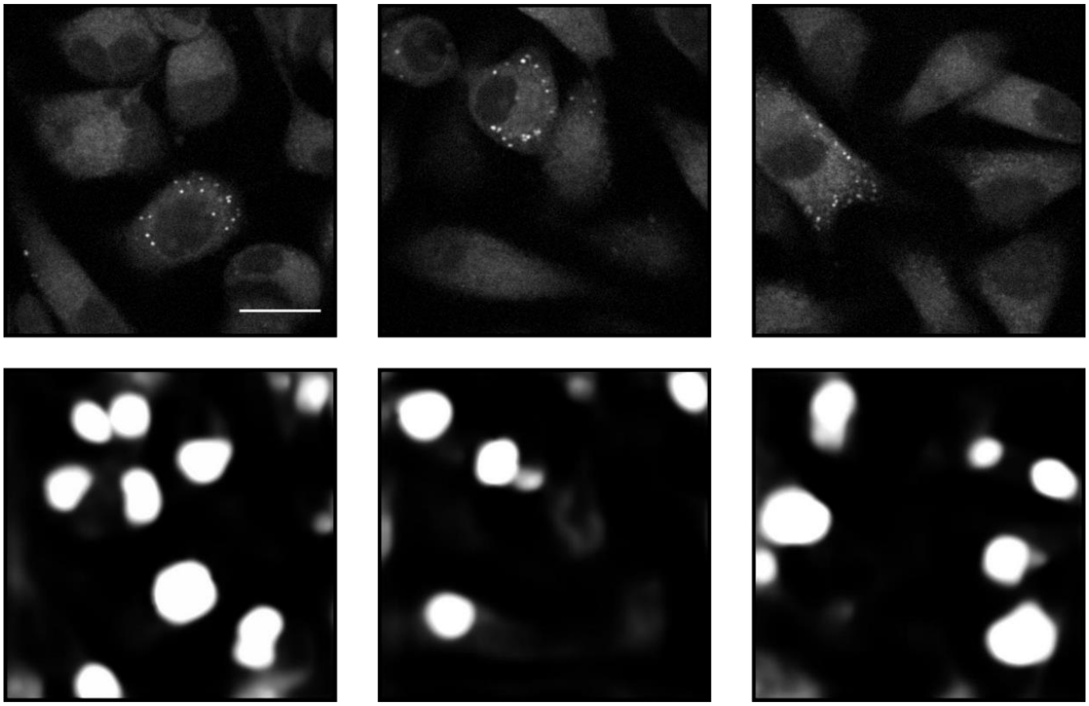


Fig. 34. Upper row: test images; lower row: prediction results. Scalebar: 20µm

The interesting thing is that, if we retest the training image again, the CNN algorithm can find the nucleus that was ignored by the thresholding algorithm due to relatively weaker illumination as shown in Fig. 35. It means that the CNN has learned features of nucleus other than the mere brightness.

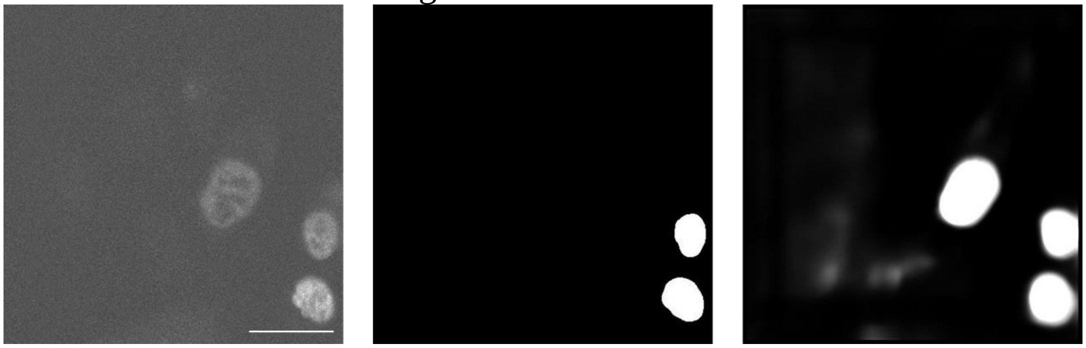


Fig. 35. Fluorescent image(left). Mask by thresholding algorithm(middle). Prediction results(right). Scalebar: 20µm

The figure above gives us an interesting hint: although the label was attained by the thresholding algorithm, the CNN is capable of excelling the thresholding algorithm in some respects, learning features that were ignored.

Therefore, a nature analogy comes that, given the concentration maps acquired by the spectral unmixing algorithms like VCA and ICA, if we train the CNN with the concentration maps, it is likely that CNN can learn some information that is ignored by the unmixing algorithms, e.g., the spatial information. Here comes the second application of CNN in analyzing SRS images, i.e., spectral unmixing of the raw SRS images.

5.3. Regression on the concentration map

The training we have done before is a process of classification, the label for a particular pixel is a discrete number corresponding to the class that the pixel could belong to., e.g., 0 means water, 1 means cytoplasm, and 2 means nucleus. For the concentration map, however, the pixel values, ranging from 0-255 for RGB three channels, indicate the relative concentration of the corresponding components. For example, a pixel with 200 of the red channel values (lipids channel), 50 of the green channel values (protein channel), and 5 of water channel values (water channel) mean that around 80% lipids, 20% protein and barely any water exist in the current pixel. The number of combinations of the three concentrations is equal to 256^3. Hence it would be impractical to list all the combination cases and do classification. Therefore, instead of doing classification, we do regression. The main difference between regression and classification is that the former classifies the pixels to discrete classes, while the latter predicts nearly continuous values based on the input. Therefore, the mean squared error (MSE) was calculated between the output prediction value of CNN and the ground truth value provided by VCA during the training process. It is also worthy to note that the activation function of the output layer was removed; in this way, one can measure the MSE without non-linear distortion generated by the activation function.

Multiple models were trained respectively for each component. The training datasets and the prediction results for lipids were shown in Fig. 36.

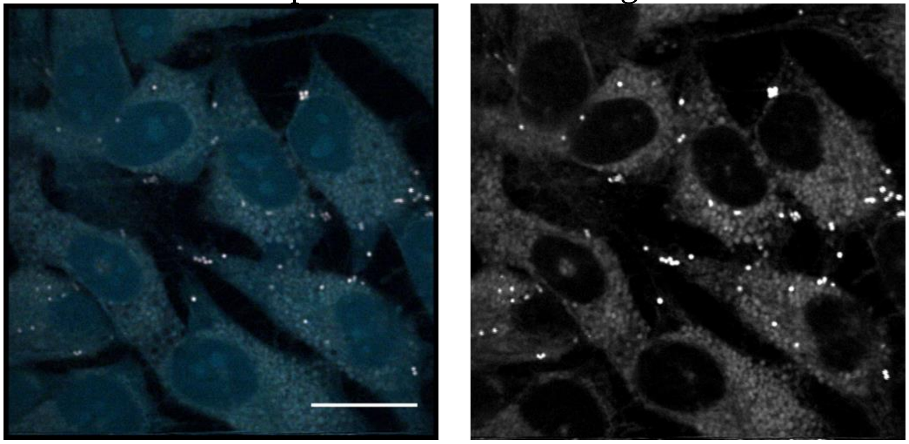


Fig. 36. Training datasets: (left) composite from images @2853cm^{-1}, @2903cm^{-1}, and @3006cm^{-1} as red, green, and blue. (right) lipids profile attained by VCA. Scalebar: 20μm

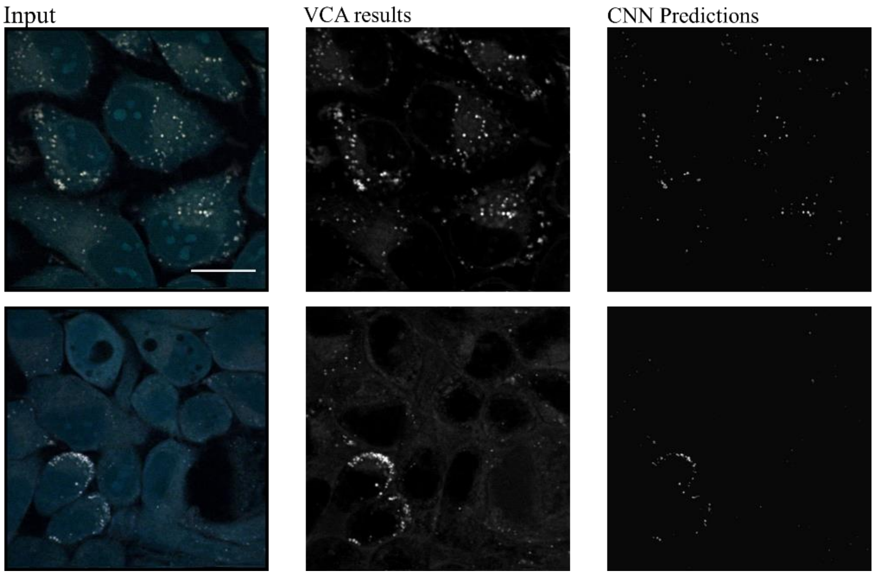


Fig. 37. Input images and the corresponding lipids concentration map attained by VCA and the prediction results of CNN. Scalebar: 20µm

From the figure above, we can find that CNN tends to pick up only high-signal lipids droplets. The pixels with lower lipids concentration, however, were oppressed and predicted as background for some reasons.

The same approach was applied to predict protein. The training dataset and the prediction results were shown in Fig. 38. We can also see a considerable resemblance between the VCA protein concentration map and the predicted profile.

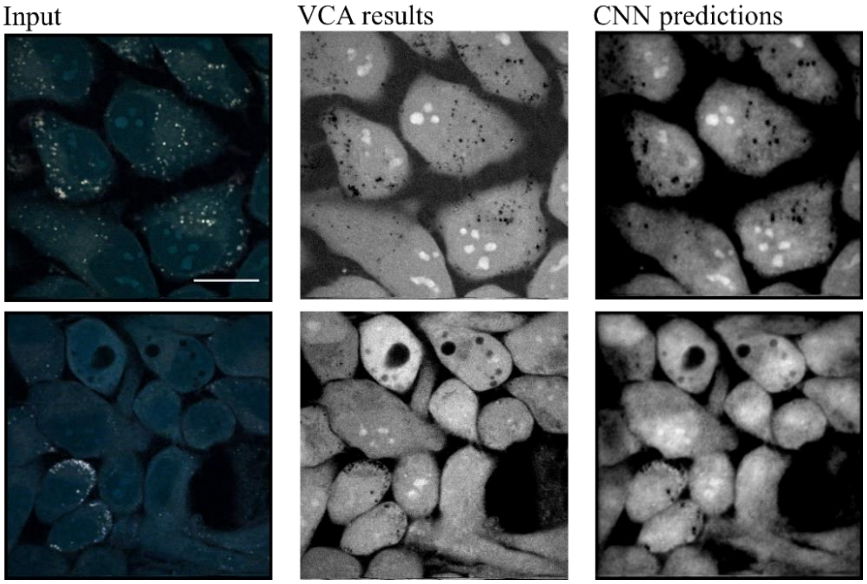


Fig. 38. Input images and the corresponding protein concentration map attained by VCA and the predictions of CNN. Scalebar: 20µm

The interesting point is that CNN was trained on the SRS image datasets with only 3 spectral channels. VCA, however, used 91 spectral channels to acquire the concentration map. If we apply VCA only on 3 spectral channels as we trained CNN, there will be an obvious deterioration in the quality of the concentration maps. The results of VCA on the 3-spectral-channel were shown in Fig. 39.

We can compare the 3-channel VCA protein profile, Fig. 39(right) and the 3-channel CNN prediction results above (upper right). It is evident that although VCA can manage to get precise profiles of lipids with only 3-channel images thanks to the high signal intensity of lipids, it cannot give a satisfying concentration map of protein. The trained CNN, however, can provide us with a protein concentration map with much higher SNR.

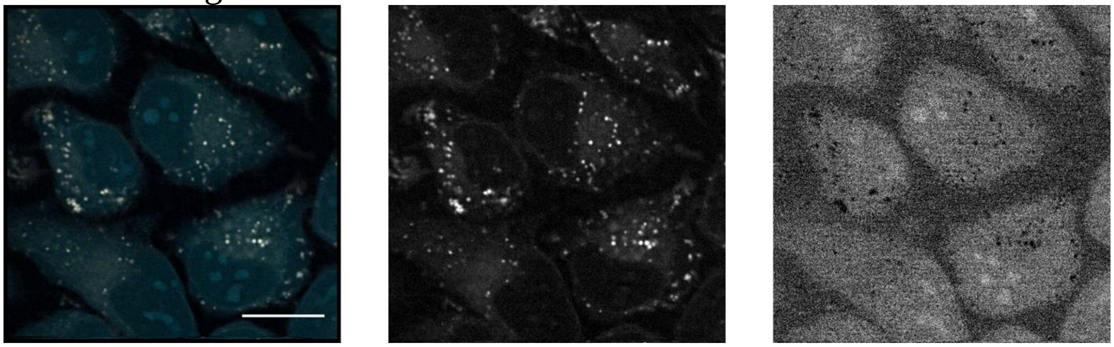


Fig. 39. (left) composite SRS images @2853cm^{-1}, @2903cm^{-1}, and @3006cm^{-1} as red, green, and blue. (middle) Lipids profile acquired by VCA. (right) Protein profile obtained by VCA. Scalebar: 20μm

The implication here is, by only taking three images at specific wavelengths and then inputting it to a pre-trained CNN model, we can get the corresponding distribution of lipids and protein with acceptable accuracy. In this way, we could dramatically decrease the time spent on data acquisition while acquiring reasonable concentration maps.

Furthermore, we could even use only one spectral channel image and the corresponding concentration map to train the CNN model. The SRS image at 2935cm^{-1} was selected as the input image. In this case, the spectral unmixing algorithms would not work since there was only one image at hand. The CNN model, however, could give us acceptable concentration profiles without much deterioration of accuracy from the 3-channel case (Fig. 40).

The validation accuracy for the 3-channel input and 1-channel input after 40 epochs is shown in Table 1. The comparison of 1-channel and 3-channel prediction results of lipids are shown in Fig. 41.

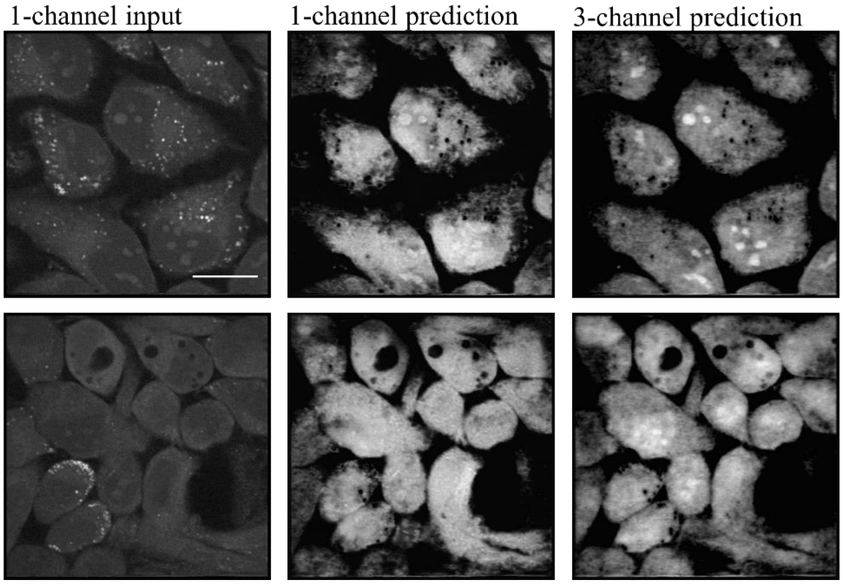


Fig. 40. (left)1-channel input SRS image @2935cm^{-1}, (middle)prediction protein of 1-channel model and (right) prediction protein of 3 channel model. Scalebar: 20µm

Table 1. The validation accuracy of the CNN model after training for 40 epochs

Validation accuracy	3-channel input	1-channel input
Protein	86.69%	84.07%
Lipids	99.93%	99.91%

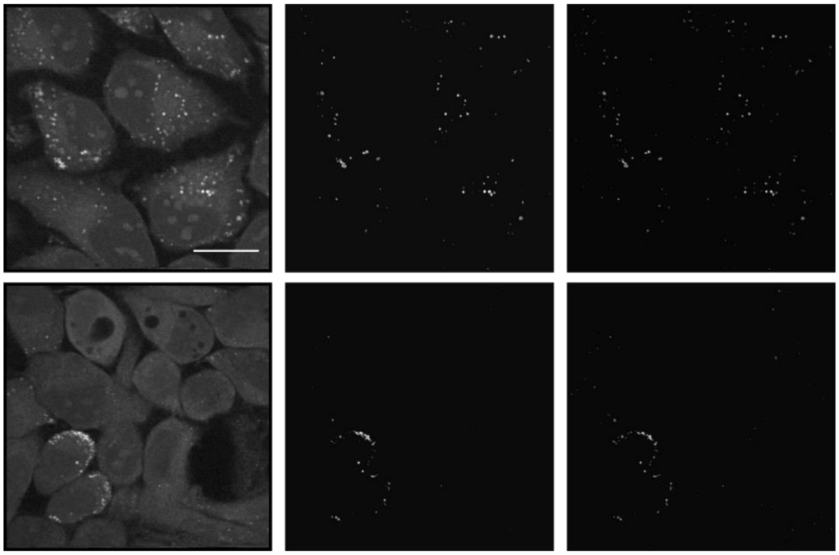


Fig. 41. (left)1-channel input SRS image @2935cm^{-1}, prediction results of 1-channel model (middle) and 3-channel model(right) for lipids concentration. Scalebar: 20µm

The merged image from 91-channel VCA, 3-channel CNN predictions, and 1-channel CNN predictions are shown in Fig. 42.

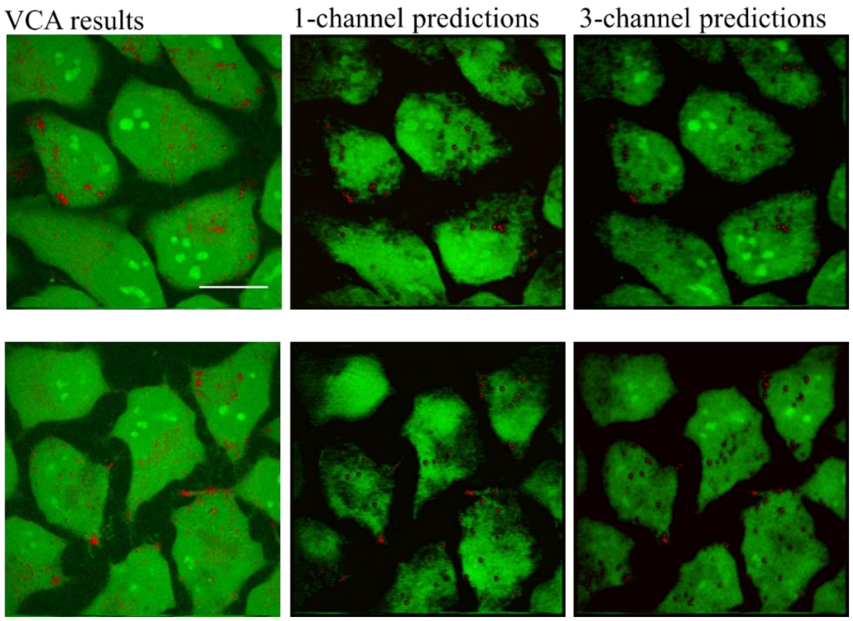


Fig. 42. (left) VCA results, (middle) prediction results of 1-channel model and (right) 3-channel model for lipids concentration. Scalebar: 20µm

As mentioned before, we cannot see apparent changes in the CNN prediction results for both protein and lipids when we decreased the spectral channel number from 3 to 1. It means that for the CNN model. The 3-channel SRS image may include a lot of informative redundancy, and it would work well with just 1-channel SRS image. This finding is encouraging because we could incorporate a pre-trained model into the SRS hardware, and then the pre-trained CNN model could enable a quick and real-time decomposition of SRS images. It may not be as much as accurate as of the spectral unmixing algorithms given 91-channel hyperspectral SRS data cube, but it could provide an entirely acceptable decomposition results based on only one SRS image.

5.4. Summary

As so far, we have covered the utilization of CNN model in the segmentation of cell structures and spectral unmixing of SRS data.

For the segmentation part, we talked about masking techniques from traditional thresholding to recently emerged CNN architecture. The masking techniques enable us to build cell libraries at the single-cell level and reveal important information of cells (see appendix). Furthermore, fluorescent imaging experiments were done to help us get more accurate label images with fewer training costs.

For the spectral unmixing part, we used the label images from spectral unmixing algorithms, i.e., VCA to train the model. We also slightly modified the model to make regression rather than classification. The trained CNN model gave an excellent performance even with only one SRS image as input. In the future, this approach could enable the recognition of lipids and protein in a real-time and accurate manner. Although the combination of CNN and other algorithms and biological techniques seems promising, a lot of work still could be done in this field. For example, due to the limitation of GPU storage, only three spectral dimensions were incorporated, but in our SRS hyperspectral data cube, there are 91 spectral channels. The performance could get better if more channels being integrated. There are a lot of techniques that can help us tackle this problem: compressing of input images, modification of network structure, parallel computing on multiple GPUs, and so on. The thing that needs to be considered is how to strike a balance between computational resources, information redundancy, and prediction precision.

Besides, what we have done now is known as semantic segmentation. It is the pixel-wise segmentation and attributes every pixel to a predetermined class. The algorithm cannot know, however, how to recognize different instances, which means it cannot tell us how many cells or nuclei exist in an image and where the boundary between two overlapped cells is. By incorporating the instance location and add another loss function to penalize the distance of the predicted location form the true location, we could achieve the instance segmentation.

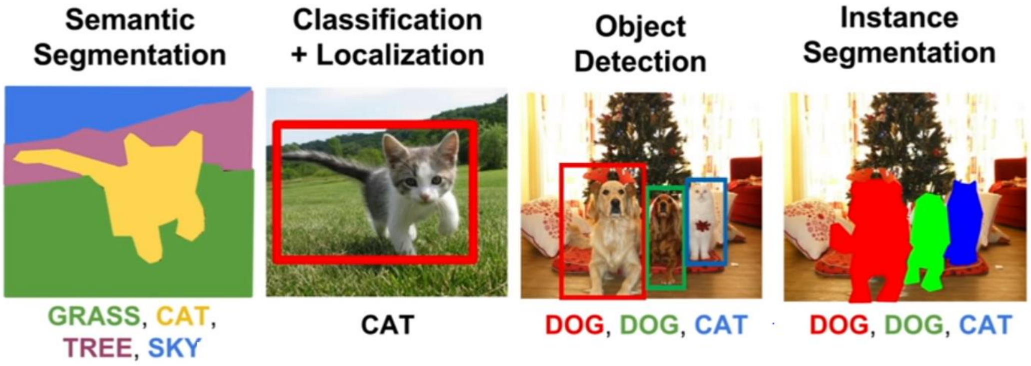


Fig. 43. Explanation of different image recognition tasks (CC0 public domain)

Furthermore, denoising with CNN algorithm is another promising area[16]. This group used SRS images with low SNR (created with a low laser power) and the ones with high SNR (created with high laser power) to train the U-net network. After being trained, the model can help denoise the new images and gave better performance than the traditional algorithms.

6. Discrimination of spectral nuances with PCA

Spectral unmixing algorithms can help us capture the conspicuous differences in spectra, and then with the extracted spectra, we can decompose the SRS data to get the corresponding concentration maps. Some finer nuances of the spectra, however, could also reveal important information. One of our experiment, for example, was designed to observe cells intake of a kind of unsaturated lipids known as EPA (Eicosapentaenoic acid). The major spectral difference of EPA and the saturated lipids we have looked at before is that EPA has a much higher peak at 3010cm^{-1}, as shown in Fig. 44.

In our experiment, Hela cells were cultured in an EPA-abundant environment and then were observed under SRS microscope. There should be both EPA and also original saturated lipids in the cell. However, if we were to use VCA or ICA to decompose the image, only spectral differences between lipids, protein, and water can be discriminated, but the spectral nuances between EPA and saturated lipids would be overshadowed.

To further understand the distribution of EPA in cells, we need to take a closer look at the lipids pixels. Therefore, first VCA was used to select the first 200 “purest” lipids pixels from the EPA group and the control group (cultured without EPA). The spectra of the pixels are shown in Fig. 44.

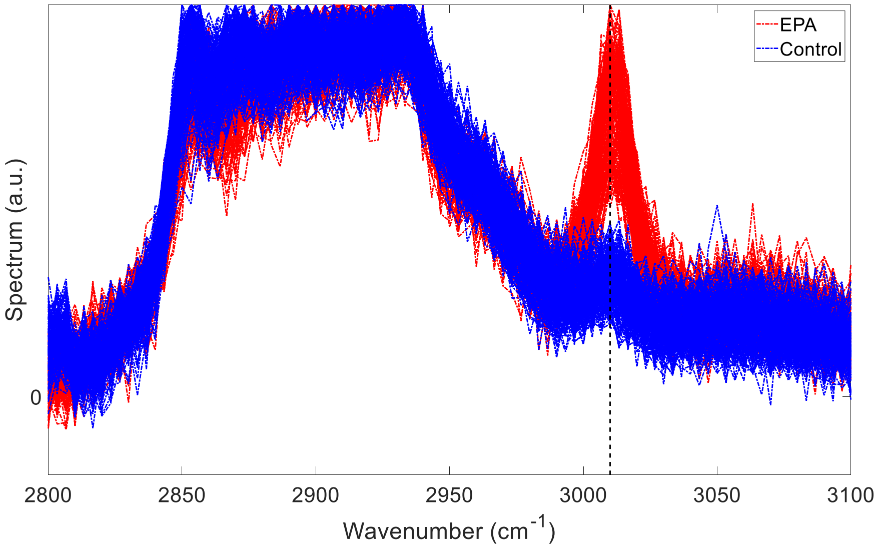


Fig. 44. Spectra of the first 200 “purest” lipids pixels selected by VCA for the EPA group and the control group.

From the figure above we can see that the for EPA group, lipids spectra have an obvious higher peak than the control group at 3010cm^{-1}. For the rest, however, the spectra of the two groups overlap a lot. Therefore, there is some similar spectral information between the two groups but also the discrepancy. In this case, PCA could be used to cluster the high-correlated data groups and make them uncorrelated with each other.

According to this intuitive thought, PCA was applied to the SRS data above. The first two loading vectors were plotted in Fig. 45. The loading vector of the first principal component (PC1) seemingly stands for the saturated lipids profile that is shared by both groups, and the second one (PC2) appears to represent the characteristic EPA peak aforementioned.

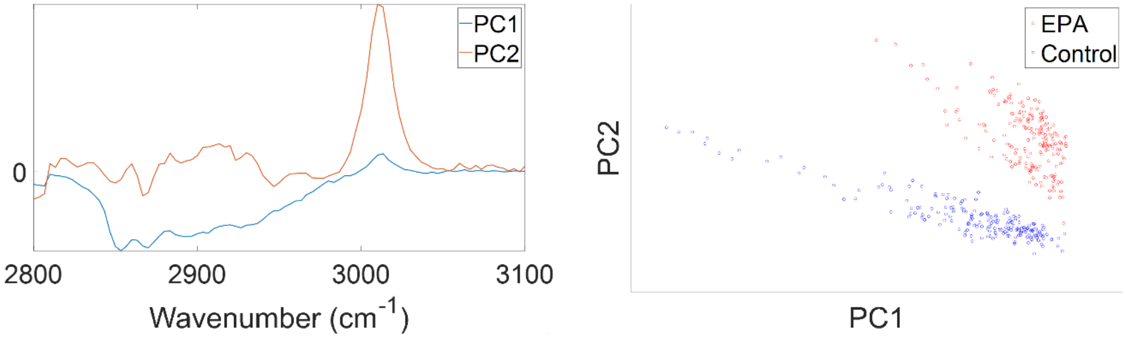


Fig. 45. The first two loading vectors of PCA (left) and the projection of two groups in principal space(right)

And we can find a clear separation of the two groups in the principal space. The EPA group has higher PC2, the seemingly EPA peak component, than the control group given the same PC1 amount.

It is worthy of mentioning that it is possible to get a negative loading vector as that of PC1. It is because both the positive one and the opposite would give the same variance. Meanwhile, the first principal component PC1 accounts for 73.84% variance of the original data, while the second one PC2 only accounts for 11.43%.

By applying PCA to the EPA group and the control group, we can somehow know how much “the EPA peak component” one lipid pixel has by looking at the amount of PC2. As we mentioned before, for the EPA group itself, there should also be different kinds of lipids in the cells: the original saturated ones as indicated by PC1 and the EPA ones. Therefore, if we also apply PCA to only EPA group, a similar result is expected.

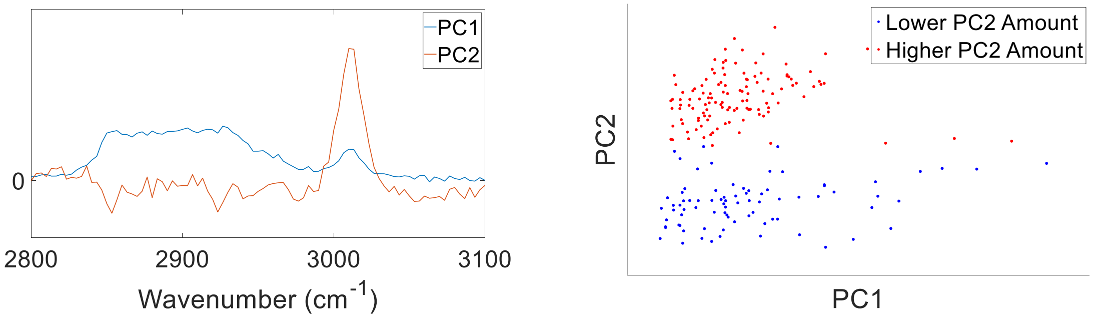


Fig. 46. The first two loading vectors(left) and the scatterplot of the EPA group(right)

It turned out that we attained similar loading vectors. The first principal component PC1 accounts for 54.87% variance while the second one PC2 only accounts for 7.78%.

Because there is no clear boundary between "Lower PC2 Amount" group and the "Higher PC2 Amount" group, I just plotted the first 120 pixels out of 200 as the "Higher PC2 Amount" group (red) and the rest 80 as the "Lower PC2 Amount" (blue). Still, we can see some separation between the two groups, although not as same clear as the one between the EPA and the control group.

The spectra of the low-PC2 group and the high-PC2 group also show the corresponding discrepancy: We can see in Fig. 47 that the former has relatively lower EPA peak than the latter.

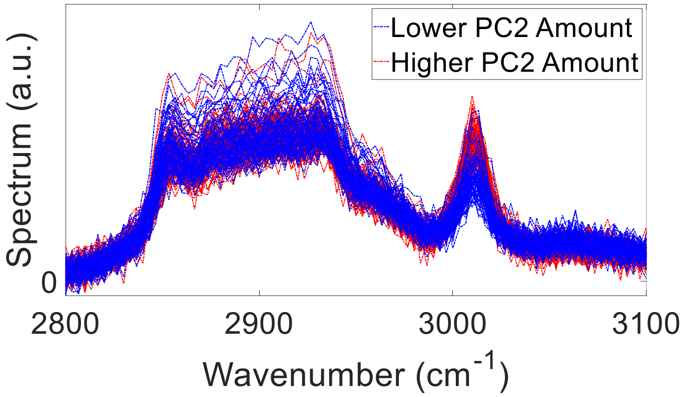


Fig. 47. Spectra of the low-EPA group and the high-EPA group

However, another important thing we want to do is to observe the EPA distribution in the cells. Although VCA could give us the first N purest lipids, it cannot know on earth how many lipids pixels in the cell. Therefore, it can only provide us an uncompleted distribution map as shown in Fig. 48.

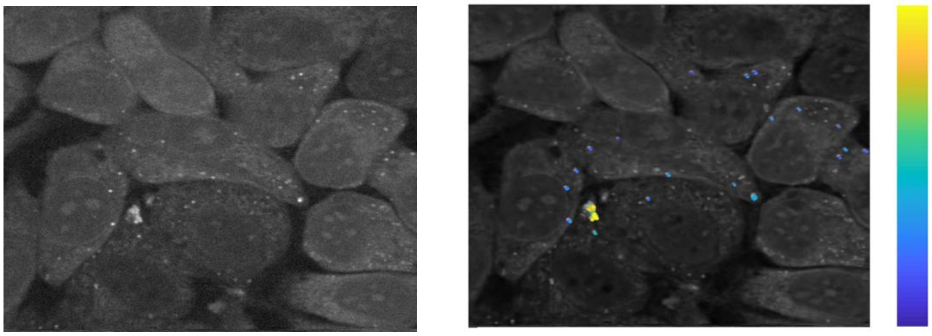


Fig. 48. (left)raw SRS image @2935cm^{-1}. (right)limited lipids droplets selected by VCA, the warmer the color, the higher the concentration of PC2 (the EPA peak).

To address these two problems, we can use the pre-trained CNN model aforementioned to predict the lipids pixels and then apply PCA to these pixels. The CNN model is supposed to find a non-linear mapping between the raw SRS images and the corresponding lipids and protein concentration maps. By training the CNN regression model with the VCA concentration map, it could predict concentration maps from the new input images. From the previous results, we can find that the CNN model tended to suppress the pixels with lower lipids concentration, which could improve the selectivity of lipids pixels by removing less purer ones.

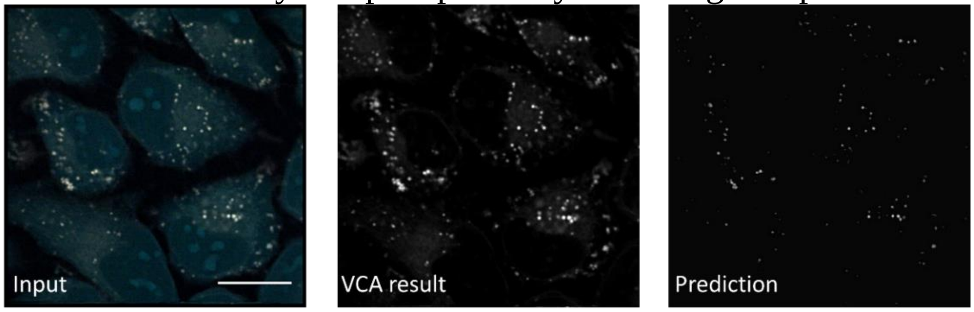


Fig. 49. Lipids prediction given by CNN is darker than the VCA result. Scalebar: 20µm

Then we apply the Otsu method aforementioned to threshold the predicted image and to get the corresponding masks of lipids. Due to the high contrast of lipids pixel and background, a reasonable mask can be obtained without applying Gaussian filters. Please see Appendix for the specific introduction of this thresholding approach.

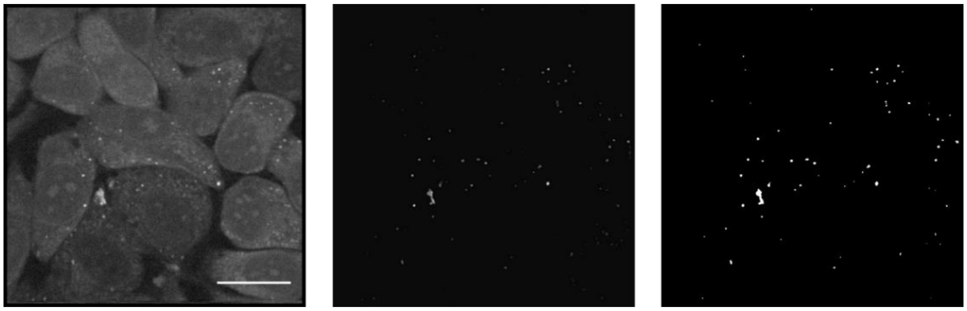


Fig. 50. (left)Raw SRS image. (middle) CNN prediction. (right) the mask given by Otsu. Scalebar: 20µm

Now we have more predicted lipids pixels than VCA has given before. By applying PCA to these pixels and composite the PC2 (EPA peak component) concentration map and the raw SRS image, we can get the concentration distribution of PC2 Fig. 51.

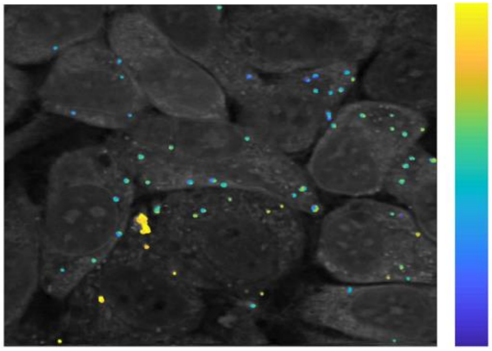


Fig. 51. PC2 concentration map

If we compare Fig. 51, and Fig. 48(right) we can find that there is a consistency between the distribution of PC2 concentration. But CNN provided us with more plausible pixels of lipids droplets and thus enabled a more comprehensive understanding of EPA intake.

The PC spectra and the scatterplot of CNN-predicted pixels (Fig. 52) also show similarities with the VCA-selected ones (Fig. 46).

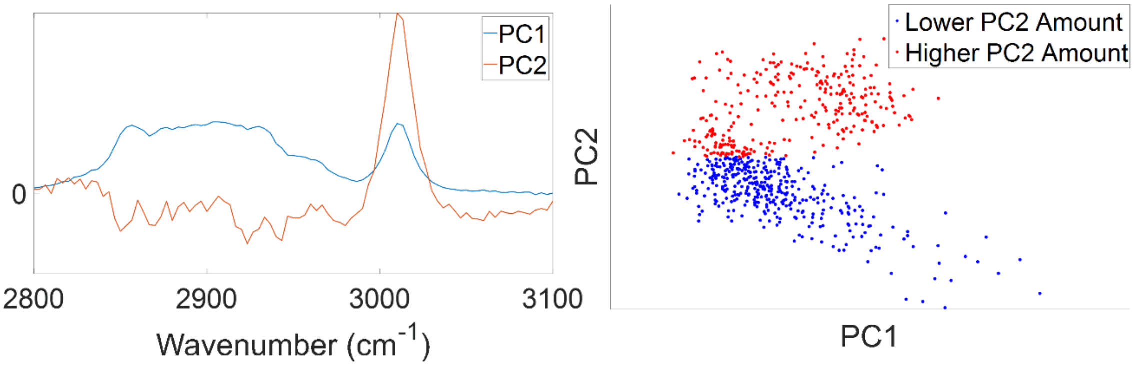


Fig. 52. CNN predicted lipids pixels (N=665): (left) PC spectra (right) projection in principal space, with the mean value of PC2 as boundary

It is also worthy of mentioning that, except for non-linear supervised learning algorithm like CNN, the supervised algorithms that learn features in a linear way, such as support vector machine (SVM) can also be trained to extract the lipids pixels. SVM strives to find a liner boundary that maximizes the minimum distance between the boundary and the neighboring data points. The logic behind the applicability of SVM for SRS data is that we assume that lipids-abundant pixels and other pixels are linearly separable in the 91-dimension space (91 is the number of spectral channels).

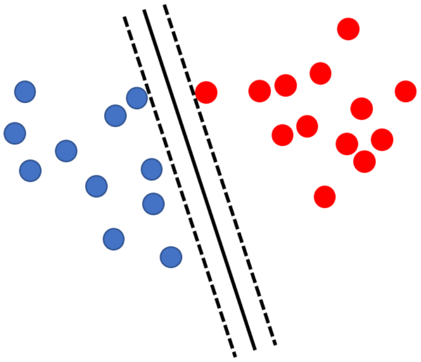


Fig. 53. A simple illustration of SVM: the algorithm tries to find a boundary that maximizes the distances (margin) from both sides.

If we first train the SVM with VCA selected pixels of lipids and protein and then input the rest of pixels into the trained SVM model for prediction, we can also get reasonable classification results as shown in Fig. 54.

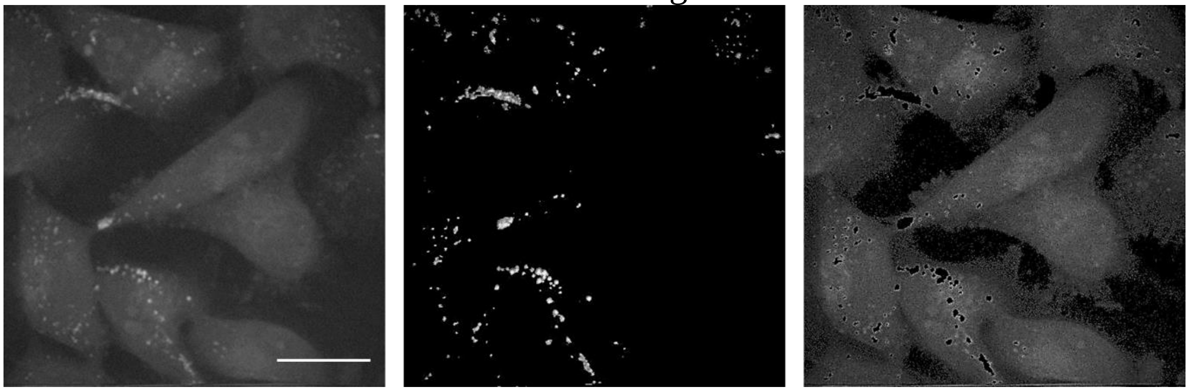


Fig. 54. (left)raw SRS image. (b)pixels classified as lipids. (d)pixels classified as the protein. Scalebar: 20μm

Later I applied PCA to retrieve spectral nuances from the SVM-predicted lipids. The concentration profile of PC2 (EPA peak component) is shown in Fig. 55.

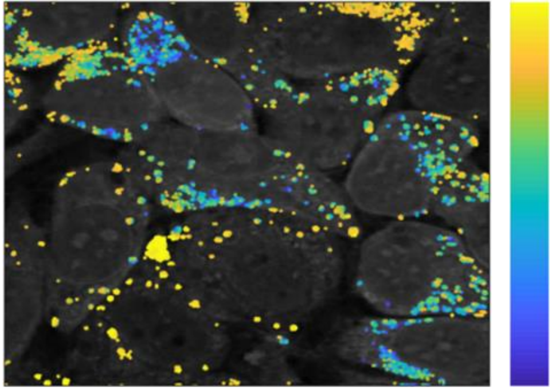


Fig. 55. concentration profile of PC2 for SVM-classified lipids

The spectra of the first two principal components and the distribution of pixels at the principal space are shown in Fig. 56.

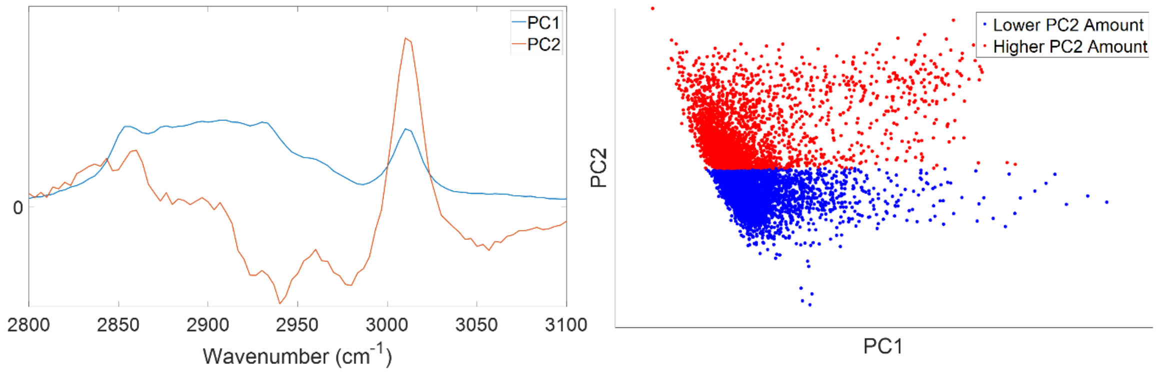


Fig. 56. (left) Spectra of the first two principal components (right) distribution of pixels at the principal space, with the mean value of PC2 as boundary

We can find that the PC2 spectra of the SVM-predicted pixels are somehow different from the previous VCA and CNN ones: there are two obvious peaks around 2940cm^{-1} and 2980cm^{-1}. A possible explanation is that a lot of pixels of the protein-lipids-water mixture were also classified as pure lipids pixels by SVM, rendering the spectrum of PC2 influenced by water and protein. If we check the average concentration of lipids, protein, and water for the selected lipids pixels of the three methods, we can find that the SVM-predicted ones have a higher protein and water concentration and lower lipids concentration compared to the rest two. The normalized VCA concentration is shown in Table 2.

Table 2. Normalized mean concentration for the pixels selected by the three methods.

	VCA	CNN	SVM
Lipids	90.61%	67.63%	29.69%
Protein	5.30%	18.83%	36.51%
Water	4.08%	13.54%	33.81%

Furthermore, if we plot the spectra of pixels in the order of descending lipids concentration, we can get a figure as shown in Fig. 57. We can find that the shape of the spectrum is shifting to protein when the concentration of the lipids decreases. The characteristic peak of protein around 2040cm^{-1} could explain the peak at the same location of PC2 in Fig. 56(left).

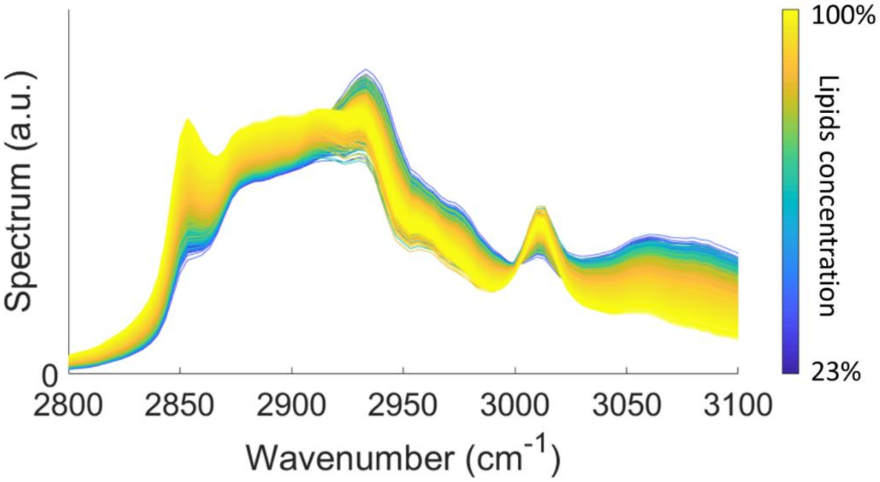


Fig. 57. The spectra of lipids of SVM-predicted pixels

If we put the three concentration profiles of PC2 of the VCA-selected, CNN-predicted and SVM-predicted lipids together as shown in Fig. 58, we can find a certain consistency between the three. While VCA gave us the limited number of lipids pixels, CNN predicted more pixels as lipids and SVM classified the largest number of pixels as lipids.

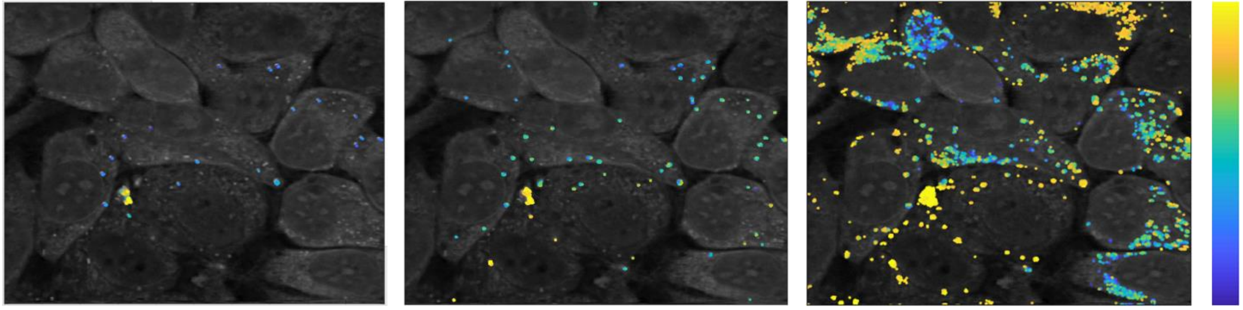


Fig. 58. The concentration profile of PC2 for the pixels of lipids given by (a) VCA, N=200, (b)CNN, N=665, and (c)SVM, N=7900. N is the number of supposed lipids pixels.

In conclusion, by using PCA to decompose the lipids pixels, we could discriminate finer spectral nuances like the spectral differences between unsaturated lipids droplets (EPA) and the saturated ones. Since VCA cannot know exactly how many lipids pixels in an image dataset, CNN and SVM were trained by VCA results to separate lipids pixels in a non-linear and linear manner respectively. Although all the three algorithms gave a different number of lipids predicted, there is a certain consistency among the concentration maps of PC2 (EPA peak component) for all the three methods. We can also have a clear and direct impression of the EPA intake situation with PCA decomposition.

7. Conclusion

I first introduced the principle of SRS microscopy and experimental setup in Chapter 2: when we illuminate the molecule of interest with two trains of lasers at different frequencies, the energy will transfer from the one with higher frequency to the one with lower frequency. When the beating frequency matches the vibrational frequency, the transfer will come to climax. Besides, the transferred energy is proportional to the concentration of the molecule. Therefore, if we tune one laser over a continuous range of wavelength, we will attain the vibrational spectra of the molecule, which will give us a signature of the molecule. This is the basic idea of SRS microscopy.

In chapter 3 we first stated the mathematical definition of the spectral unmixing problem: we need to recover the vibrational spectrum and the concentration profile of different components from their linearly mixed observations, i.e., the raw SRS images. The existing algorithms that could address the problem, namely, MCR, PCA, ICA, and VCA, were introduced. Since the equation to be solved is ill-posed, all of the algorithms above try to add constraints to narrow the solution space. MCR emphasizes on physical restraints, e.g., non-negativity of the spectrum and concentration; PCA assumes that the hidden variables should be uncorrelated and ranked by their contribution to the total variance. ICA supposes that the source signals should be independent with each other hence tries to find the projection that maximizes the independence. Instead of exploiting the statistical information like PCA and ICA, VCA locates the purest pixels and then picks up the corresponding spectrum directly.

All the algorithms aforementioned have their pros and cons, and comprehensive evaluation was made in Chapter 4. It turned out that PCA only decorrelated the data and cannot give us physically interpretable results. ICA, however, could give a reasonable spectrum. VCA worked similarly as manual extraction but would not attain pure spectra if there were no pure pixels in the images. For MCR, although the results were reasonable, due to the existence of local minimums and random initials, it tended to give different results every time. Finally, I compared the performance of ICA, VCA, and MCR (with the initial value fixed) in different SNR situation. The finding here is that VCA performed strikingly well for the extraction of lipids spectrum even under low SNR but much less well for protein compared to MCR and ICA. One reason could be that VCA tried to find individual pixels with high signal intensity just like the pixels of lipids, while ICA and MCR did the global optimization thus performed more stable when SNR decreased. A possible combination would be extracting protein with ICA while unmixing lipids with VCA.

Chapter 5 introduces a CNN architecture known as U-net into the spectral unmixing and the segmentation of cell structures. Rather than decomposing the data according to pre-determined features, CNN tries to learn the features. I first manually labeled the different structures, e.g., cytoplasm and nucleus and made training images from 3 raw SRS images at different spectral channels. Later I trained the U-net with the label and the training images, and it achieved a good classification of cell structures. Furthermore, I found that if I incorporate the component map of protein unmixed by VCA into the training images, better recognition of nucleolus would be achieved. It may be because the protein map made the nucleolus stand out and thus strengthened its features for learning. Later I used the fluorescent image of the stained nuclei to make the label images, and in this way, we could get training datasets more easily and accurately.

Later I trained the U-net with the component maps from VCA and then let U-net make a regressive prediction on the given raw SRS images. It turned out that although CNN may not give a concentration map as accurate as 91-channel VCA, it outperformed VCA with quite a small number of channels since some features ignored by VCA, e.g., spatial information, was learned by CNN. Therefore, a pre-trained CNN model could enable separation of components in a fast or even real-time manner.

Chapter 6 talked about how we can further discriminate finer spectral differences: I used the EPA-cultured samples and the control groups to show how to classify EPA (unsaturated lipids) and saturated lipids by their spectral features. PCA was used here, and I reserved the first two principal components—the first one corresponded to saturated lipids, and the second one represented the feature peak of EPA. In this way, we can plot the concentration map of EPA. Furthermore, I trained CNN and SVM with VCA results to predict more lipids pixels.

In conclusion, the thesis tried to analyze the SRS images from the two respects: spectral information and spatial information. A proper combination and exploitation of the two respects could reveal us abundant information that we wanted to know about the cells. How to strike the proper balance between the accuracy, the computational costs and the speed of analysis under a specific situation would be much of interest in future research.

References

[1] Cheng, J. X. & Xie, X. S. Vibrational spectroscopic imaging of living systems: An emerging platform for biology and medicine. *Science* **350**, aaa8870 (2015).

[2] https://sites.google.com/site/ozekibp/research

[3] Ozeki, Y., Umemura, W., Otsuka, Y., Satoh, S., Hashimoto, H., Sumimura, K., Nishizawa, N., Fukui, K. & Itoh, K. High-speed molecular spectral imaging of tissue with stimulated Raman scattering. *Nat. Photonics* **6**, 845–851 (2012).

[4] de Juan, A. & Tauler, R. Multivariate curve resolution (MCR) from 2000: Progress in concepts and applications. *Crit. Rev. Anal. Chem.* **36**, 163–176 (2006).

[5] Bartholomew, D. J. Principal components analysis. *Int. Encycl. Educ.* **2**, 374–377 (2010).

[6] Fabian Pedregosa, Gael Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand ¨ Thirion, Olivier Grisel, Mathieu Blondel,Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot and Edouard Duchesnay. Scikit-learn: Machine learning in Python. *J. Mach. Learn. Res.* **12**, 2825-2830 (2011).

[7] A. Hyvärinen, Juha Karhunen, Erkki Oja, Independent component analysis. Wiley (2001).

[8] Nascimento, J. M. P. & Dias, J. M. B. Vertex component analysis: A fast algorithm to unmix hyperspectral data, *IEEE Trans. Geosci. Remote Sens.* **43**, 898–910 (2005).

[9] Berry, M. W., Browne, M., Langville, A. N., Pauca, V. P. & Plemmons, R. J. Algorithms and applications for approximate nonnegative matrix factorization. *Comput. Stat. Data Anal.* **52**, 155–173 (2007).

[10] Otsu, N., Threshold selection method from gray-level histograms. *IEEE Trans. Syst. Man. Cybern.* **SMC-9**, 62–66 (1979).

[11] Krizhevsky, A., Sutskever, I. & Hinton, G. E. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems.* 1097–1105 (2012).

[12] Simonyan, K. & Zisserman, A. Very deep convolutional networks for large-scale image recognition. arXiv:1409.1556 (2014).

[13] Ronneberger, O., Fischer, P. & Brox, T. U-net: Convolutional networks for biomedical image segmentation. *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)* **9351**, 234–241 (2015).

[14] http://brainiac2.mit.edu/isbi_challenge/

[15] https://cellprofiler.org/

[16] Manifold, B., Thomas, E., Francis, A. T., Hill, A. H. & Fu, D. Denoising of stimulated Raman scattering microscopy images via deep learning. *Biomed. Opt. Express* **10**, 3860–3874 (2019).

[17] Suzuki, Y., Kobayashi, K., Wakisaka, Y., Deng, D., Tanaka, S., Huang, C.-J., Lei, C., Sun, C.-W., Liu, H., Fujiwaki, Y., Lee, S., Isozaki, A., Kasai, Y., Hayakawa, T., Sakuma, S., Arai, F., Koizumi, K., Tezuka, H., Inaba, M., Hiraki, K., Ito, T., Hase, M., Matsusaka, S., Shiba, K., Suga, K., Nishikawa, M., Jona, M., Yatomi, Y., Yalikun, Y., Tanaka, Y., Sugimura, T., Nitta, N., Goda, K. & Ozeki, Y. Label-free chemical imaging flow cytometry by high-speed multicolor stimulated Raman scattering. *Proc. Natl. Acad. Sci. U.S.A.* **116**, 15842–15848 (2019).

[18] Lee, S. Y. Plastic bacteria? Progress and prospects for polyhydroxyalkanoate production in bacteria. *Trends Biotechnol.* **14**, 431–438 (1996).

List of presentations

First author presentation

[1] 劉寒沁, 浅井卓也, 寿景文, 小関泰之「誘導ラマン散乱顕微鏡のためのスペクトル分離法の性能比較」応用物理学会秋季学術講演会、20p-PB6-8、名古屋国際会議場、2018年9月20日。

Presentation and publication as a coauthor

[2] 浅井卓也、劉寒沁、小関泰之、林智広、佐藤伸一、中村浩之「ホウ素クラスター化合物の生細胞への取り込みの誘導ラマンイメージング」第66回応用物理学会春季学術講演会、9a-W641-6、2019年3月9日。

[3] Suzuki, Y., Kobayashi, K., Wakisaka, Y., Deng, D., Tanaka, S., Huang, C.-J., Lei, C., Sun, C.-W., Liu, H., Fujiwaki, Y., Lee, S., Isozaki, A., Kasai, Y., Hayakawa, T., Sakuma, S., Arai, F., Koizumi, K., Tezuka, H., Inaba, M., Hiraki, K., Ito, T., Hase, M., Matsusaka, S., Shiba, K., Suga, K., Nishikawa, M., Jona, M., Yatomi, Y., Yalikun, Y., Tanaka, Y., Sugimura, T., Nitta, N., Goda, K. & Ozeki, Y. Label-free chemical imaging flow cytometry by high-speed multicolor stimulated Raman scattering. *Proc. Natl. Acad. Sci. U.S.A.* **116**, 15842–15848 (2019).

Appendix

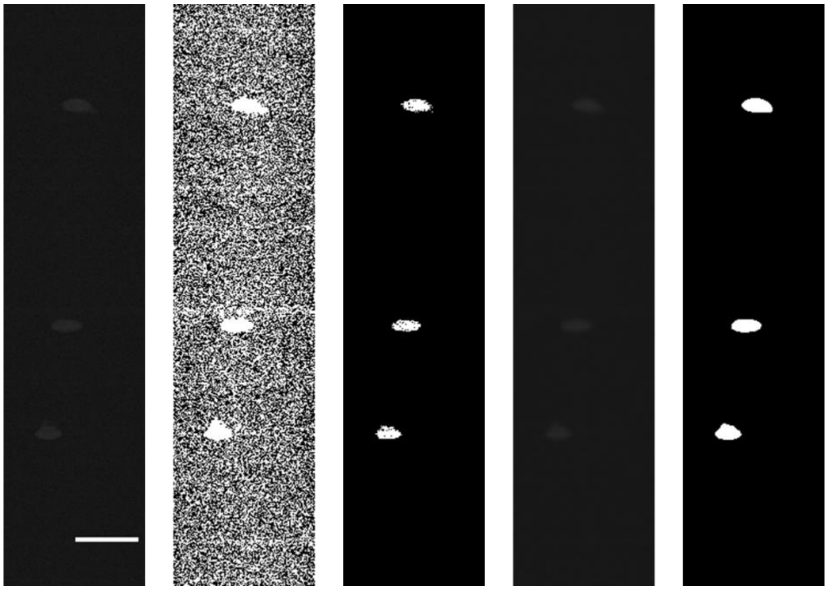
Cell masking serves as a way of extracting cell from the background. It can help us build a library for individual cells: by cell masking, information for a single cell can be retrieved afterward. There are various ways of cell masking. An intuitive way may be the thresholding—first select a proper threshold value, then set the pixel values over the threshold to 255 while pixel values under the threshold to 0, In this way hopefully we can attain reasonable masks.

A commonly used global thresholding algorithm is known as the Otsu method[10]. It is a histogram-based algorithm that tries to minimize the intra-class variance, i.e., the data bins in the same category should have the minimum variances.

However, direct application of the Otsu method could give a poor performance due to the noise, as shown in Fig. 59(b). The reason why Otsu cannot separate cells from the background is that the extensive distribution of the brightness of background pixels overwhelms the signal pixels (Fig. 60(a)). By appropriately adjusting the threshold value, e.g., multiplying it by a constant 1.5 in our case (Fig. 60(a)), we could mask the cells with higher accuracy.

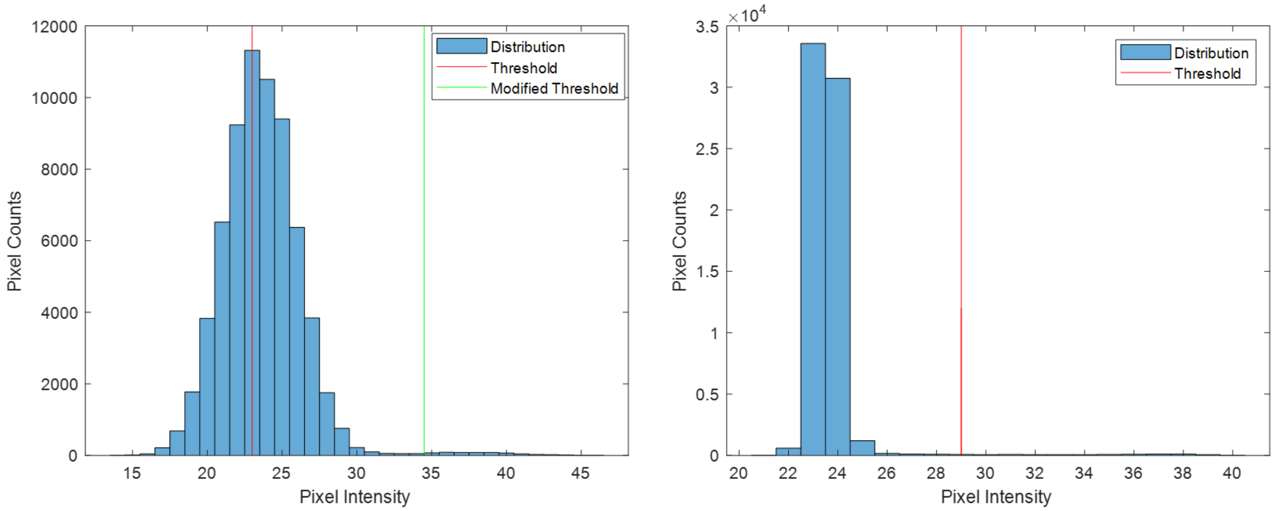
However, there are two major drawbacks in this approach: i) manual setting of a parameter would be time-consuming. ii) Due to the uneven intracellular illumination, there are still some pixels within cells being classified as background, as shown in Fig. 59(c).

A simple preprocessing method could help us avoid the two problems, i.e., adding a spatial filter, such as the Gaussian filter to the image before applying the Otsu method. The Gaussian filter averages the current pixel with its neighboring pixels by Gaussian weight. In this way, the image gets smoother (Fig. 59(d)), and both the cell pixels and background pixels get more concentrated in the histogram (Fig. 60(b)): the background pixels tend to concentrate from 22 to 26, while the cell pixels concentrate from 35 to 39. The concentration of data bins from different classes help the Otsu algorithm easily find out a more accurate threshold. In one experiment, we used the approach to mask four types of blood cells (Fig. 61)



(a) (b) (c) (d) (e)

Fig. 59. (a) Jurkat cells taken at 3040cm^{-1}. (b) Results of the Otsu method without modifying the threshold. (c) Results of the Otsu method with the threshold value timed by 1.5. (d) Jurkat cells after a Gaussian filter with sigma = 2 pixels. (e) Results of the Otsu method after the image being processed by the Gaussian filter. Scalebar: 20µm



(a) (b)

Fig. 60. Histogram of the raw image without (a) and with (b) using the Gaussian filter

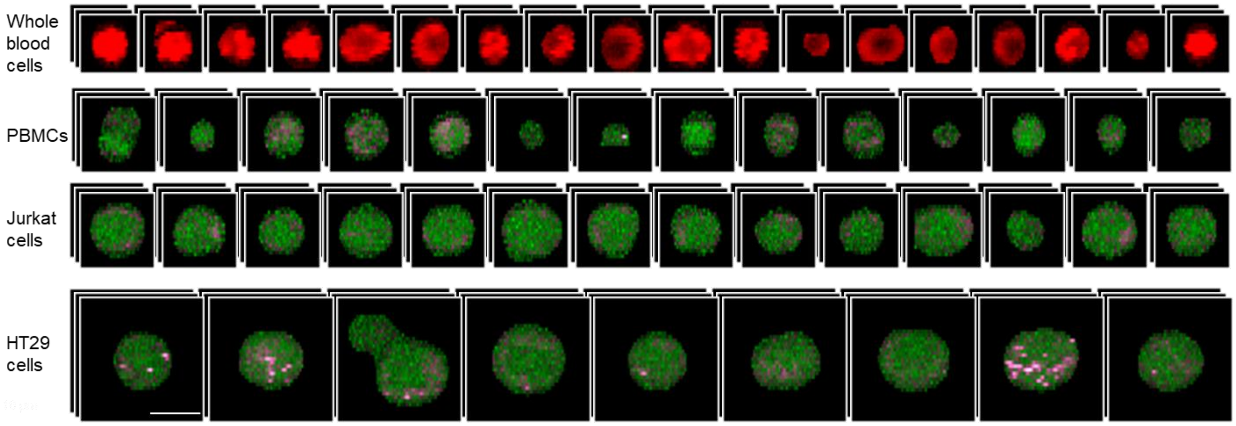


Fig. 61. Cell libraries for four types of blood cells[17], constructed by applying masks attained by Gaussian filter and Otsu method. Scalebar: 10μm

The approach of masking aforementioned was applied to mask cells observed by SRS flow cytometry system, and the results are shown in the figure above.

Although using Gaussian filter does help us get clearer masks, another problem arises: Because that the Gaussian filter tends to average pixels and its neighboring pixels, it makes masks slightly rounder and larger than they are. This issue would not be quite severe if the dimension of cells is way larger than a single pixel, e.g., a Jurkat cell usually takes up hundreds of pixels. But the Gaussian filter would distort the mask to a relatively large content if the size of the cell is comparable to a single pixel (around tens of pixels, for example). Besides, if there are two small cells close to each other, it is likely that after the Gaussian filter, they will connect and then make the Otsu method misrepresent the two cells as one big cell.

In another experiment, this kind of issue happened: when we tried to mask PHA bacteria, a kind of bacteria that is promising in producing biodegradable plastics[18], The size of PHA bacteria is far less than the normal cells we have seen before. Therefore, to avoid the dilation effect brought by the Gaussian filter, some other denoising methods are necessary.

Fortunately, since all the pixels within the bacteria have a similar SRS spectrum, we can use the spectral unmixing algorithms aforementioned to extract the component and then use the component to represent the bacteria itself. In this way, we can get a cleaner representation of raw images for the following Otsu method. It turned out that the component-profile-based masking outperformed the Gaussian-filter-based masking in the following aspects: 1) distinguish closely located bacteria. 2) Retrieve the bacteria with extremely small areas, which could have been overwhelmed by the noisy neighboring pixels after the Gaussian filter. 3) Prevent the dilation of masks.

It is worthy of mentioning that, however, when there are multiple components in a cell, we cannot use only one component to make masks because pixels of other components would get missed. That is why we cannot directly apply this approach to cells that have various components.

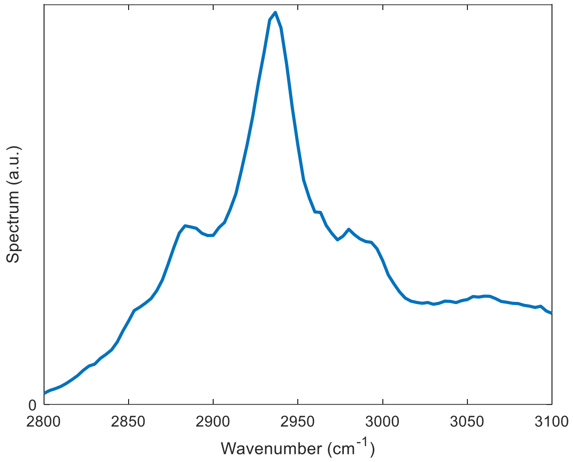


Fig. 62. PHA Spectrum extracted by VCA

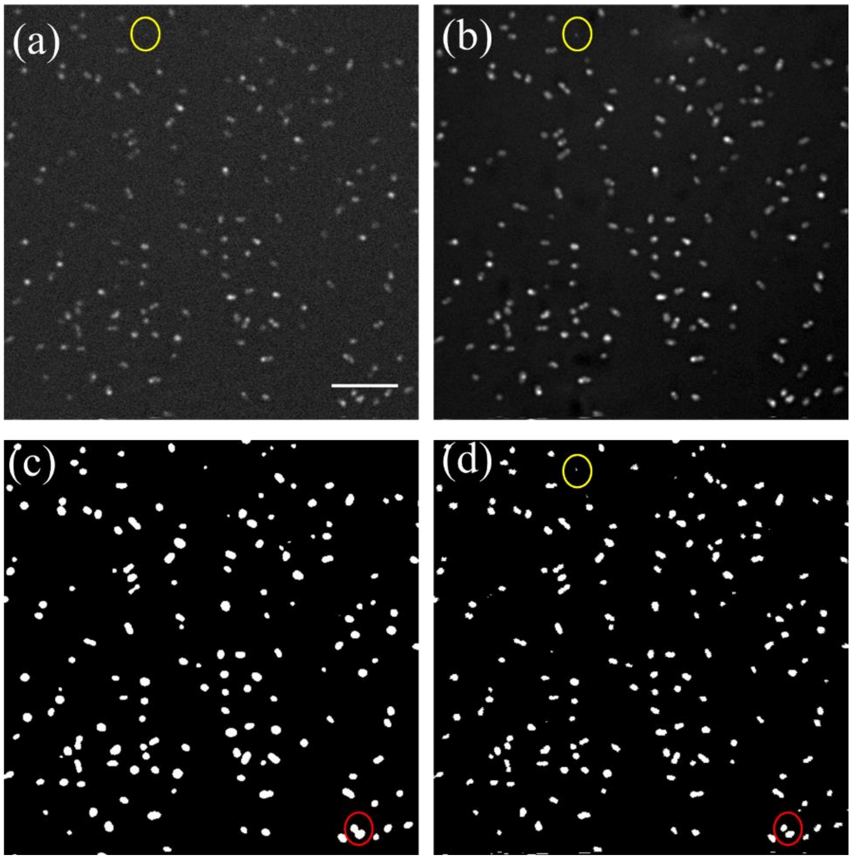
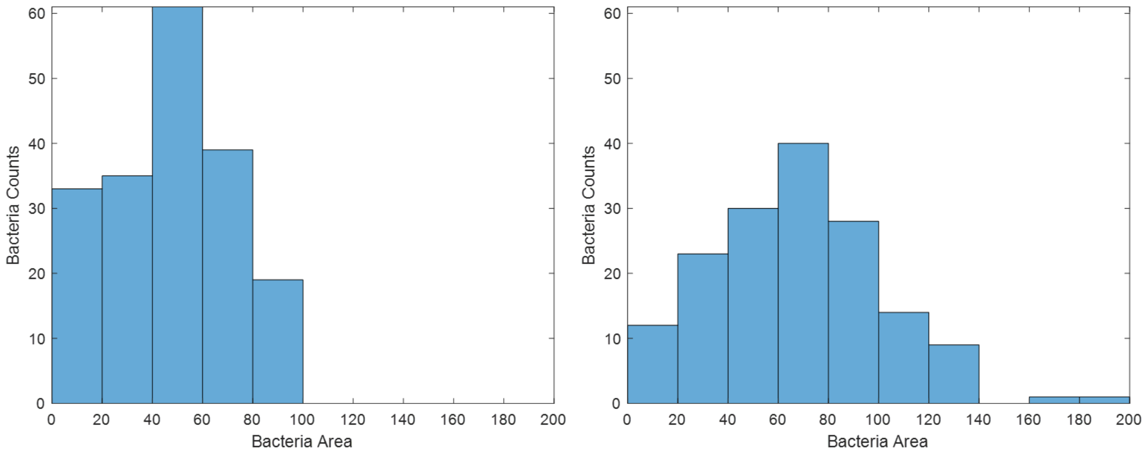


Fig. 63. (a) SRS image of PHA bacteria at 2932cm^{-1} (b) Concentration map after VCA. (c) Masks from the Gaussian filter image with sigma=2. (d) Masks from the concentration map. Red marks circled two bacteria misjudged as one big bacterium from the Gaussian filter image while being separated correctly from the concentration map. Yellow marks circled the signal pixels ignored after Gaussian filter while remaining distinguished in the mask from the concentration map. Scalebar: 20µm

The histogram of mask areas from the Gaussian-filter-based mask image (b) and concentration-map-based mask image (a) are shown in Fig. 64. One can see that compared to the Gaussian one, the bacteria mask from the concentration map are more concentrated and on average have smaller areas.



(a) (b)

Fig. 64. Histogram of mask areas from the concentration-map-based mask image (a), and the Gaussian-filter-based mask image (b).

By applying the concentration-map-based cell masking, we could analyze the PHA bacteria on the level of a single bacterium, e.g., the distributions of PHA signal intensity under different culture environments are shown in Fig. 65. Since the intensity of bacteria signals reflect their capacity of producing plastics, we can evaluate which kind of cultivation environments would optimize the production by looking at the histogram distribution.

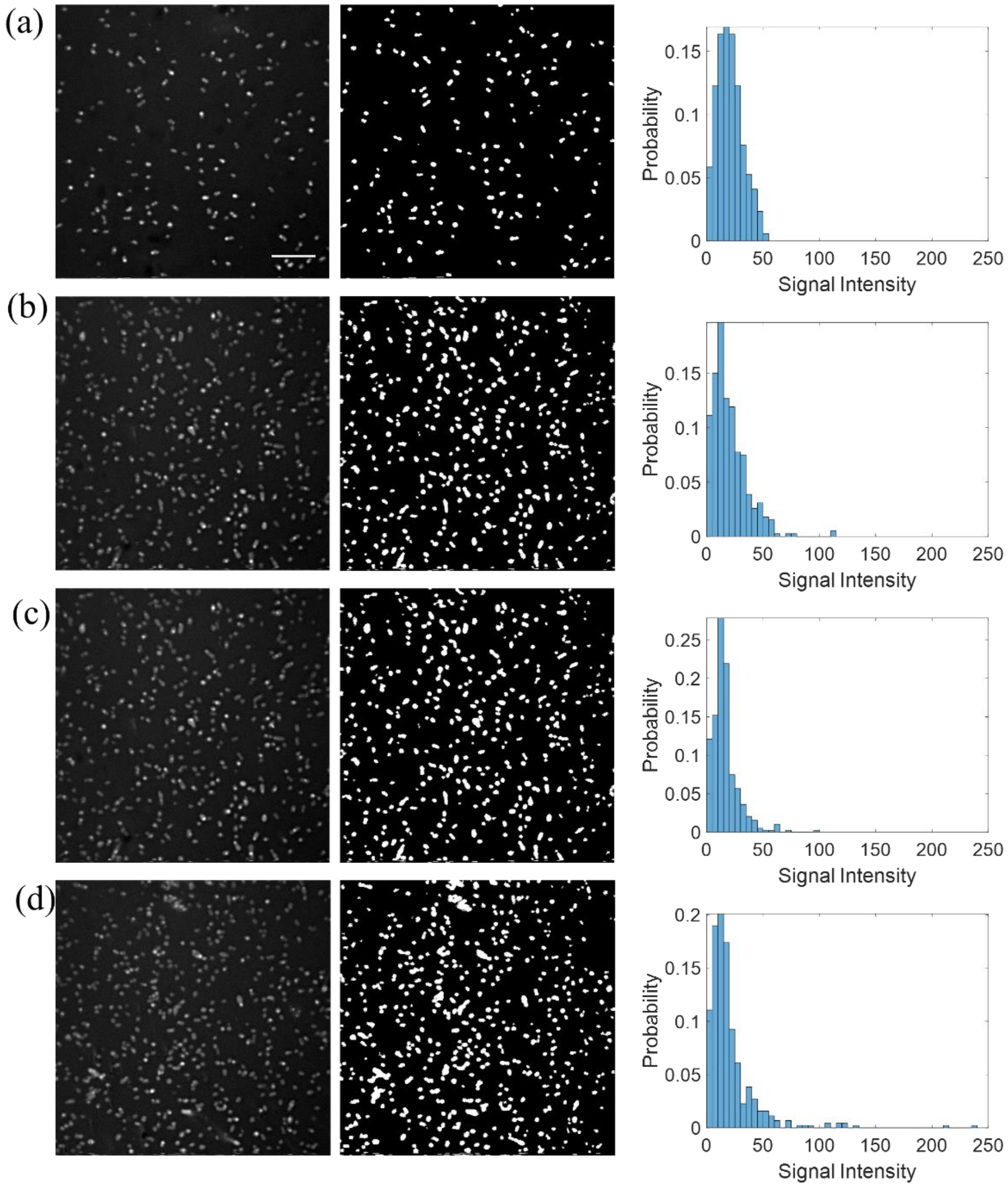


Fig. 65. PHA under different cultivation conditions (a), (b), (c), and (d) shows different distributions. Scalebar: 20µm

In conclusion, when the structures of cells or bacteria are relatively simple, the mask can be reasonably made with preprocessing steps and the Otsu method. When the size of cells is far larger than a single pixel, Gaussian filter could be applied before the Otsu method; while if the size of cells is small and only one component exists in the cell body, in order to avoid the distortion brought by the Gaussian filter, spectral unmixing algorithm could be used as the preprocessing step.