Classifying Raman spectra of extracellular vesicles based on convolutional neural networks for prostate cancer detection

Wooje Lee1 | Aufried T.M. Lenferink2 | Cees Otto2 | Herman L. Offerhaus1

1 Optical Sciences, MESA+ Institute for Nanotechnology, University of Twente, Enschede, The Netherlands
2 Medical Cell BioPhysics, Technical Medical Centre, University of Twente, Enschede, The Netherlands

Correspondence
Herman L. Offerhaus, Optical Sciences, MESA+ Institute for Nanotechnology, University of Twente, Enschede, The Netherlands.
Email: h.l.offerhaus@utwente.nl

Funding information
Stichting voor de Technische Wetenschappen, Grant/Award Number: 14197

Abstract
Since early 2000s, machine learning algorithms have been widely used in many research and industrial fields, most prominently in computer vision. Lately, many fields of study have tried to use these automated methods, and there are several reports from the field of spectroscopy. In this study, we demonstrate a classification model based on machine learning to classify Raman spectra. We obtained Raman spectra from extracellular vesicles (EVs) to find tumor derived EVs. The convolutional neural network (CNN) was trained on preprocessed Raman data and raw Raman data. We compare the result from CNN with results from principal component analysis that is widely used among in spectroscopy. The new model classifies EVs with an accuracy of >90%. Moreover, the new model based on CNN is also suitable for classifying the raw Raman data directly without preprocessing with a minimum accuracy of 93%.

KEYWORDS
Cancer biomarker, convolutional neural network, extracellular vesicles, machine learning, Raman spectroscopy

1 INTRODUCTION

Raman spectroscopy allows to extract chemical information from a sample without labeling.[1,2] When we obtain the vibrational spectrum from a pure sample like toluene, ethanol, or silicon, we can readily identify the chemical contents. In real life, samples are unlikely to include only one pure chemical component.[3–5] Especially in clinical or biological applications, samples include many different types of molecules indicative of their function or cellular origin.[3,6–10] Thus, we obtain very complex Raman spectra and analyzing spectral data requires an extended effort.[4,5,11–13]

To analyze these deeply convoluted data, principal component analysis (PCA) has commonly been employed. PCA is mostly used to reduce the dimension of the data and to make a prediction model. PCA calculates principal components of the data and projects given data onto a newly generated coordinate system.[14,15] PCA shows optimal performance if the spectral data are linearly correlated to their chemical content. Spontaneous Raman data are linear in first approximation but practical Raman spectra are unlikely to be only linear because they contain background and other features that disturb the scaling. Therefore signal processing is generally a prerequisite, and this can bias the result of PCA.[12,13]

The main sources of background signal in a biological sample are (a) autofluorescence[4,5,12,13] and (b) suspension solutions and sample container, such as phosphate-buffered saline or cell culture medium. These solutions can contribute peaks or bands to the Raman spectrum.[12] Background signals strongly affect the result of the
analysis so that we need to perform a rigorous background removal to avoid biasing of analysis result. However, removing background and noise, partially associated with the background, is challenging, because the background subtraction itself can induce peak shifting, leaning, and rejection of small peaks and distortion. It is difficult to distinguish background noise from useful Raman spectrum in a complex sample. Because the background removal is both essential and a source of errors, using a type of data analysis that can automatically handle raw data directly is useful. We can use computing power where it is most useful, and we can eliminate a human source of artifacts.

In our previous research, we classified Raman spectra obtained from extracellular vesicles (EVs) to detect prostate cancer without labeling.\(^{[16]}\) EVs are small spherical particles (about 30 nm ~ 1 \(\mu\)m in diameter) secreted by mammalian cells.\(^{[17,18]}\) Cells in general will contain different molecules depending on their function or disease state. Thus, EVs originating from diseased cells are likely to have a different cellular content compared with EVs derived from healthy cells.\(^{[17,19–21]}\) Thus, finding the cellular origin of EVs gives us an insight and potential route to disease diagnosis.

We obtained Raman spectra from four types of extracellular EVs; two out of four are blood product derived EVs, and the other two are prostate cancer cell line derived EVs (also known as tumor derived EVs [tdEVs]). In this research, we are aiming to discriminate spectral differences of Raman spectra of tdEVs from blood derived EVs as a disease biomarker. Healthy people should not have tdEVs or only a very low presence of tdEVs in their body fluids.\(^{[22]}\) Hematopoietic cell-derived EVs are always present in healthy people and patients. Hence, we aim to distinguish between hematopoietic cell-derived EVs and tdEVs using an algorithmic analysis technique. We do not aim to distinguish between healthy prostate derived EVs and prostate tumor derived EVs. Although that is interesting in itself to reveal the spectral changes, it is not the clinically relevant distinction for diagnosis.

In the prior study, we demonstrated that PCA (after preprocessing) can classify EVs depending on their cellular origin. With PCA, we were able to classify the Raman spectra with 95% of accuracy using the spectral fingerprint region (400-1800 cm\(^{-1}\)).\(^{[16]}\) Although the method showed good results, it classified data that were preprocessed and that model is not suitable for raw data. Here, we propose a prediction model based on a machine learning (ML) algorithm. ML is widely used in computer vision,\(^{[25,24]}\) voice recognition, and voice synthesis,\(^{[25,26]}\) and there were some attempts to use ML in the field of spectroscopy.\(^{[27–31]}\) We demonstrate an ML-based prediction model to classify the Raman spectra of EVs without data preprocessing. Specifically, we use a convolutional neural network (CNN) to build our prediction model.

The CNN\(^{[32–34]}\) is inspired by the mammalian brain. Layers in the brain extract features from input before this information enters the deeper areas of brain for further processing. It was revealed by Hubel and Wiesel in the 1950s who showed that feature extraction is used in pattern recognition tasks.\(^{[35]}\) In 1998, LeCun et al. applied a feature extractor in their pioneer convolutional networks.\(^{[36]}\) This neural network algorithm has the feature extracting layer known as the convolution layer prior to the feed forward neural network. In contrast to an artificial neural network, the convolution layer in CNN allows the model to extract small details and to be trained on the extracted details of the input data, which improves its prediction accuracy.\(^{[36,37]}\) Since then, CNNs have been widely used for image recognition or image classification, and there were several trials to use artificial intelligence algorithm to study Raman spectral data\(^{[27–31,38]}\) as well as different types of spectroscopic data.\(^{[39–42]}\) In this study, we suggest a platform for Raman signature classification of EVs based on CNN. The classification performed in this article is aimed at finding the spectral differences between prostate cancer derived EVs and blood cell derived EVs, because the latter are the clinically relevant background of the measurement. The platform approach will provide an automated and robust classification tool for a potential prostate cancer biomarker detection.

## 2 | EXPERIMENTAL

### 2.1 | Sample preparation

We prepared four different EV subtypes for this study: two from blood products (red blood cells and platelets) and the other two subtypes from prostate cancer cell lines (prostate cancer cell line [PC3] and lymph node carcinoma of the prostate [LNCaP]). The red blood cell concentrate and the platelet concentrate were obtained from the blood bank, Sanquin (Amsterdam, the Netherlands). The blood products were diluted 1:1 with filtered phosphate-buffered saline followed by three times centrifugation. The supernatant was pooled to collect the separated EVs.

We used the same protocol to harvest EVs from the PC3 and LNCaP cell lines. The EVs from PC3 and LNCaP were used as a model system for prostate cancer-derived EVs. Cancer cell lines were cultured at 37°C and 5% CO\(_2\) for 48 hr. After 48 hr of cell culture, the culture medium was collected and centrifuged at 1,000xg for 30 min to get rid of undesired particles, for instance apoptotic cells and bigger EV populations. After the centrifugation, the supernatant was pooled to obtain the EVs.

Transmission electron microscopy images were taken to verify the collection of the EVs. The size distribution and concentration of the sample were assessed by nano-
tracking analysis (NS500; Nanosight, Amesbury, UK), see Figure 2. The nano-tracking analysis showed a mean size for red blood cell derived EVs as 148 ± 3.7 nm at a concentration of 0.85 × 10^8 ± 0.03 × 10^8 particles per millimeter. For platelet-derived EVs, we find 89 ± 4.6 nm and 0.42 × 10^8 ± 0.02 × 10^8 particles per millimeter. For PC3-derived EVs: 172 ± 3.7 nm and 1.00 × 10^8 ± 0.03 × 10^8 particles per millimeter. For LNCaP-derived EVs the mean is 167 ± 4.4 nm, and the concentration is 1.06 × 10^8 ± 0.05 × 10^8 particles per millimeter. More details are available in our previous work.[16]

2.2 Raman spectral signature collection

We obtained the Raman signal of EVs using a home-built confocal Raman microscope.[16,43] This provides Raman measurements and optical trapping. The Raman microscope uses a Kr⁺ laser at a wavelength of 647 nm as the excitation source. The laser is focused onto the sample through a 40X objective. The same objective is also used to collect the back-scattered photons. The signal is dispersed in a home-built spectrograph. For the collection of the EVs spectral fingerprint, we used glass slides with a small cavity. The cavity was filled with 50 μl of sample and covered by a cover glass. The excitation beam was focused in the middle of the cavity. Trapping event can be readily noticed by monitoring Rayleigh scattering. Once intensity of Rayleigh scattering is increased, we recorded 16 spectra with an exposure time of 10 s per spectrum (in total 160 s). Since we measure the sample at a fixed position for a sufficiently long time, we strongly believe that we measured multiple EVs instead of single EV. In this way, we obtained 300 spectra from the four EV subtypes (75 spectra per each subtype). Figure 3 shows averaged Raman spectra of each subtype. In the figure, blue line shows Raman spectrum after the background removal, and the red curve represents averaged raw data. The raw data is shifted for clarity. Data collection is described in detail in our previous work.[16]

2.3 CNN architecture for training on Raman spectral data

The CNN architecture that is used in this study is illustrated in Figure 1. The network has three convolution layers with a max pooling layer for each convolution layer for feature extraction. The feature extractor is followed by a fully connected network for learning on the extracted features. Output from the fully connected layers is normalized by softmax into a probability distribution that is the set of probabilities of K possible outcomes. Thus, the normalized output must be in the interval (0, 1). The networks were realized in Python (Python Software Foundation. Python Language Reference, Version 3.6.6. Available at http://www.python.org) and using Tensorflow (TensorFlow: large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org) (See Supporting Information).

Figure 4 shows a diagram of a convolution layer for 1-D input data. In our CNN architecture, the input spectral data in the moving window are convoluted with n × 1 filter(s), and the filter(s) determine(s) the size of the moving window. Next the convoluted input is activated by Leaky rectified linear unit (ReLU). Leaky ReLU is given by

$$f(x) = \begin{cases} x, & \text{if } x \geq 0 \\ ax, & \text{otherwise} \end{cases}$$

(1)

where x is the input to the neuron and the parameter a is normally smaller than 1 or zero for ReLU. After convolution, the convoluted data are down-sampled by an operation known as max pooling. The max pooling reduces the spatial dimension of the convoluted feature by selecting
the maximum value in the moving window and allows the creation of a translation-invariant feature.

The extracted feature will enter the fully connected (FC) layers. In this study, the FC network is a feed-forward neural network\textsuperscript{[44\textendash}46\textsuperscript{]} with four hidden layers. In the feed forward neural network, \( I \) inputs are propagated to the adjacent hidden layer. This process is continued in every hidden layer until the end of the FC

FIGURE 2  Size distribution and concentration of EV samples measured by nano-tracking analysis. (a), (b), (c), and (d) show nano-tracking analysis results of red blood cell-, platelet-, prostate cancer cell line- and lymph node carcinoma of the prostate-derived EVs, respectively. In set shows image of EVs taken by transmission electron microscopy. Scale bar in each panel is 500 nm. This figure is reused with permission and modified after its original work.\textsuperscript{[16]} EVs, extracellular vesicles [Colour figure can be viewed at wileyonlinelibrary.com]

FIGURE 3  The averaged Raman spectra of EVs from (a) red blood cell-, (b) platelet- (c) PC3-, and (d) LNCaP-derived EVs. In each panel, the blue line represents preprocessed data, and the red line shows raw data. The shaded area shows the standard deviation of the measurement. All the spectra are normalized between 0 and 1 [Colour figure can be viewed at wileyonlinelibrary.com]
network. If the networks have \( I \) inputs connected to the next hidden layer in which the layer has \( J \) neurons, the forward propagation can be expressed as

\[
a_j = \tanh \left( \sum_{i=1}^{I} W_{ji} x_i + b_n \right) \quad j = 1, 2, 3, ..., J,
\]

where \( W_{ji} \) is a weight between \( i \)th input and \( j \)th neuron, \( x_i \) is \( i \)th input, \( b_n \) is the bias of \( n \)th hidden layer, and \( a_j \) is the output of \( j \)th neuron in the hidden layer. The output of the hidden layer will be the input of the next hidden layer or the output of the FC network with \( K \) classes; we have four classes in this study. The output of the FC is given by

\[
y_k = \sum_{j=1}^{J} W_{kj} a_j + b_{out} \quad k = 1, 2, 3, ..., K,
\]

where \( W_{kj} \) are the weights connected to the output of FC network, \( a_j \) is the output of the previous hidden layer, \( b_{out} \) is the bias of the output layer, and \( y_k \) is the \( k \)-dimensional nonactivated output of the FC. The output will be activated by the softmax function,[24,47] Equation (4).

\[
S(y_k) = \frac{e^{y_k}}{\sum_{m=1}^{K} e^{y_m}} \quad k = 1, 2, 3, ..., K
\]

The softmax calculates the probability distribution of the event over “\( K \)” different events that sum to one. To train a model on a given input, the model calculates the distance between its prediction and given label. The distance is called cost, and the cost is calculated by the cross entropy function[47,48] written as

\[
D(S, L) = - \sum_{k=1}^{K} L_k \log(S(y_k)),
\]

where \( S \) is the probability of each class and \( L \) is the given label. Here, we used one hot encoded label that means the character label is expressed as a vector, for example “RBC-EVs” is expressed as \([0 \ 0 \ 0 \ 1]\), “platelet-EVs” is \([0 \ 1 \ 0]\), and so on. The Adam optimizer updates weights based on the cost to minimize the distance between the prediction result and the target.[49] The outcome of the cross entropy function will be closer to zero if the model is trained well. Then, the model propagates new data forward, and the new cost will be back propagated iteratively for the training. During the network training, we applied dropout to avoid weight vanishing and overfitting to the training data.[50] The dropout algorithm randomly selects 50% of neurons in each layer for every iteration.

3 | RESULT AND DISCUSSION

We performed PCA and CNN both on baseline corrected data and raw data. To make a prediction model based on PCA, the EVs’ Raman data are divided into two subsets which are training and testing sets. The PCA training set consists with 240 spectral data, and the testing set has 60 spectral data. To make the training set and testing set, the spectra are evenly selected on a random basis from four EV subtypes; we selected 15 spectra from each EV subtype to make the testing dataset. PCA is done on the training dataset, and we predict the testing set based on PCA result of the training set. The PCA-based prediction model was realized in MATLAB R2016b (Version 9.1.0, The MathWorks, Natick, MA). The PCA and CNN model were trained on three different spectral regions to find most relevant spectral area for the classification: 400–3,050 cm\(^{-1}\) (full spectrum), 400–1,800 cm\(^{-1}\) (fingerprint) and 2,700–3,050 cm\(^{-1}\) (high frequency, also known as C-H stretch region).
The aforementioned 300 Raman spectra of EVs were also used for the CNN-based prediction model. Artificial neural network models usually require large volume of data to learn more small detail and avoid over fitting to the given data. Moreover, the dimension of Raman spectrum is $1,152 \times 1$. It means that the dimension of the training data is far bigger than the number of data, and it can readily cause over fitting problem. To solve this problem, we conducted data augmentation which is a commonly used method to increase the number of training data. For the data augmentation, we generated white Gaussian noise with signal to noise ratio (SNR) of 15, 25, and 30 and added to the original signal, which was done using the additive white Gaussian noise function provided by MATLAB. Figure 5 shows an example of data augmentation done for this research. After the augmentation, we had 1,200 spectral data that include 300 original data and 900 random noise added data. Then, the Raman spectral dataset is divided into three subsets as follows: training, validation and testing dataset. We randomly selected 90 spectra from each subtype. The testing set was prepared from 50% of 90 spectra, and the other 50% became validation set. In the end, we had a training set, testing set, and validation set with 840, 180, and 180 Raman spectra, respectively. The structure of the model for this particular analysis has three sets of convolution layers followed by a max pooling layer. The feature extractor is followed by four hidden layers that have 1,000, 500, 200, and 200 neurons. The output layer has four neurons and is connected to softmax to convert output scores to a probability distribution. Prior to the network training, all the weights of the network were initialized on a random basis. We assigned the weights with Xavier initializer, which assigns weights from a Gaussian distribution.\[^{[46]}\] The initialization method keeps the variance of the weights the same in each hidden layer. CNN training time was about 10 to 70 min depending on the input dimension, and all the training was done by a graphics processing unit, which is NVIDIA GTX1080Ti.

Table 1 and 2 show the classification result of PCA-linear discriminant analysis (LDA) and PCA-quadratic discriminant analysis (QDA) on preprocessed data (Table 1) and raw data (Table 2). In Table 1, PCA-QDA shows a fairly good classification, especially in the fingerprint region. However, the results show that the PCA-based model has a high prediction/classification accuracy under certain condition; it performed well on the fingerprint region ($400–1,800 \text{cm}^{-1}$) of the background corrected Raman data. In general, however, the PCA model classified the Raman spectra of the EVs poorly in the full spectral area and in the high frequency region. Table 2 shows the result of PCA-LDA and PCA-QDA trained on untreated data. As can be seen in the table, classification accuracy of PCA on raw data is very low. The result shows that PCA is not suitable for handling the raw Raman data because PCA requires decent background/noise removal process as discussed in Section 1.

We trained the CNN model on preprocessed and raw Raman data, and Table 3 shows the prediction accuracy on both datasets. In both cases, the prediction accuracy is higher than 90%. Originally, we assumed that CNN trained on clearer signal would show a better classification accuracy because, after removing background noise.

### Table 1 Prediction accuracy of principal component analysis on preprocessed dataset

| Spectral region (cm$^{-1}$) | LDA  | QDA  |
|-----------------------------|------|------|
| 400–3,050                   | 0.650| 0.783|
| 400–1,800                   | 0.833| 0.950|
| 2,700–3,050                 | 0.733| 0.867|

Abbreviations: LDA, linear discriminant analysis; QDA, quadratic discriminant analysis.

### Table 2 Prediction accuracy of principal component analysis on raw data

| Spectral region (cm$^{-1}$) | LDA  | QDA  |
|-----------------------------|------|------|
| 400–3,050                   | 0.616| 0.683|
| 400–1,800                   | 0.616| 0.616|
| 2,700–3,050                 | 0.583| 0.600|

Abbreviations: LDA, linear discriminant analysis; QDA, quadratic discriminant analysis.
contribution, the remaining data should have cleaner EVs contribution instead of noise/fluorescent contribution. Although the model trained on preprocessed data classified the spectra with accuracy of 90.89% in full spectral area and 90.22% in fingerprint and 91.22% in high frequency, the model trained on raw Raman data shows better prediction accuracies of 95.22% in full spectral region and 96.56% in fingerprint region and 93.11% in high frequency region. We attribute this to small signal buried in the untreated spectral data, which is not clearly visible because of its low SNR.

The mean size of EVs used in this study is about 150 nm. The single particle is about 100-fold smaller than the focal volume of the Raman microscope. Thus, the solution in which EVs are suspended contributes to the Raman signal more than the trapped particles do, which leads to poor SNR of the Raman spectra of EVs, about 7 dB. At such, an SNR small spectral features are concealed by background contribution, and small peaks might be eliminated by background correction. In other words, raw data retain small spectral information that is not clearly visible in the raw spectrum because of poor SNR. This subtle information allows the CNN model to learn more details of the input signal.

We have tried identifying the most relevant spectral regions that contain most of the meaningful information for the classification. The result in Table 3 shows that every spectral segment used in this study shows high accuracy of 95.22%, 96.56%, and 93.11% in fingerprint (400–1,800 cm$^{-1}$), high frequency (2,700–3,050 cm$^{-1}$), and full spectrum (400–3,050 cm$^{-1}$), respectively. The fact that the model trained on fingerprint performed better is suggesting that the spectral fingerprint region (400–1,800 cm$^{-1}$) has more relevant information for the classification than the high frequency region (2,700–3,050 cm$^{-1}$). However, it does not imply that the spectral information in high frequency region is less important than information in the spectral fingerprint region. Whereas protein and lipid contribution are more prominent in the high frequency region, many other biomolecules contribute to the fingerprint region. The PCA model shows a similar result, namely a classification accuracy on the fingerprint and high frequency region of 95.00% and 86.67%, respectively.

### 4 | CONCLUSION

In this study, we have demonstrated that a CNN-based prediction model can be used as a classifier of Raman spectra of EVs and that the model is suitable for raw data handling. The study shows that a PCA-based prediction model can classify the spectral data by EVs’ cellular origin, but its classification ability is limited by background noise and spectral range of input signal. On the other hand, the CNN model suggested in this paper shows a better classification accuracy (>90%) on both preprocessed data and raw data. Interestingly, the model trained on raw data classifies the Raman spectra of EVs better than the model trained on preprocessed data. It suggests that the use of raw data is useful for the classification because the raw data keeps more features to learn and computing power can be saved.

### ACKNOWLEDGEMENTS

This work is part of the research program (Cancer-ID) with project number (14197) which is financed by the Netherlands Organization for Scientific Research (NWO).

### CONFLICT OF INTEREST

The authors declare no competing financial interest.

### ORCID

Wooje Lee [https://orcid.org/0000-0002-6238-2898](https://orcid.org/0000-0002-6238-2898)

Cees Otto [https://orcid.org/0000-0001-6955-4843](https://orcid.org/0000-0001-6955-4843)

### REFERENCES

[1] C. V. Raman, *Ind. J. Phys.* 1928, 2, 387.
[2] K. Kneipp, H. Kneipp, I. Itzkan, R. R. Dasari, M. S. Feld, *Chem. Rev.* 1999, 99, 2957.
[3] Y. H. Ong, M. Lim, Q. Liu, *Opt. Express* 2012, 20, 22158.
[4] R. Petry, M. Schmitt, J. Popp, *ChemPhysChem* 2003, 4, 14.
[5] S. Nie, S. R. Emory, *Science* 1997, 275, 1102.
[6] X. Song, R. D. Airan, D. R. Ariffin, A. Bar-Shir, D. K. Kadayakkara, G. Liu, A. A. Gilad, P. C. van Zijl, M. T. McMahon, J. W. Bulte, *Nat. Commun.* 2015, 6, 6719.
[7] L. A. Austin, S. Osseiran, C. L. Evans, *Analyst* 2016, 141, 476.
