Autoencoder-aided measurement of concentration from a single line of speckle

Karamehmedovi, Mirza; Sehic, Kenan; Dammann, Bernd; Suljagi, Mirza; Karamehmedovi, Emir

Published in:
Optics Express

Link to article, DOI:
10.1364/OE.27.029098

Publication date:
2019

Document Version
Publisher's PDF, also known as Version of record

Link back to DTU Orbit

Citation (APA):
Autoencoder-aided measurement of concentration from a single line of speckle

Mirza Karamehmedović,1,* © Kenan Šehić,1 Bernd Dammann,1 Mirza Suljagić,2 and Emir Karamehmedović2

1Department of Applied Mathematics and Computer Science, Technical University of Denmark, Matematiktorvet 303B, DK-2800 Kgs. Lyngby, Denmark
2Faculty of Engineering and Natural Sciences, International University of Sarajevo, Hrasnička Cesta 15, 71210 Sarajevo, Bosnia and Herzegovina

* mika@dtu.dk

Abstract: We demonstrate that a single 6mm line sample of simulated near-field speckle intensity suffices for accurate estimation of the concentration of dielectric micro-particles over a range from $10^4$ to $6 \cdot 10^6$ particles per ml. For this estimation, we analyze the speckle using both standard methods (linear principal component analysis, support vector machine (SVM)) and a neural network, in the form of a sparse stacked autoencoder (SSAE) with a softmax classifier or with an SVM. Using an SSAE with SVM, we classify line speckle samples according to particle concentration with an average accuracy of over 78%, with other methods close behind.

© 2019 Optical Society of America under the terms of the OSA Open Access Publishing Agreement

1. Introduction

Transmission of coherent light through a large collection of randomly distributed small particles typically results in the formation of a speckle pattern, see Fig. 1. There is a substantial literature linking the statistics of the speckle pattern with the size, refractive index and concentration of the scattering particles, or more generally with the properties of the phase screen generating the speckle; see, e.g., [1–9] and the references therein. While there are many alternative optical methods for the estimation of properties of particle ensembles [10,11], in this paper we focus on speckle measurement. Speckle analysis presents a non-invasive and potentially fast and accurate technique for determination of concentration of different living cell types used in various diagnostic procedures in medicine and research protocols in biomedical and biotech industry. Current methods for prokaryotic and eukaryotic cell counting protocols used in diagnostics require pathogen isolation, time consuming microscopy or expensive spectrometry instrumentation together with a trained professional [12–14].

Fig. 1. Formation of speckle in the transmission of coherent light through an aerosol. Experimental speckle, green-laser illumination at 532 nm.
Speckle-based methods typically rely on the mean intensity or on the “size” of the measured speckle. “Speckle size” is connected with the variance of the speckle intensity and is often defined to be some measure of width of the central prominent peak occurring in the autocorrelation function of the speckle image. As an example of a theoretical link between the speckle intensity and the sample concentration, Dainty [2, p. 38] gives the relation
\[
\langle I \rangle = N \langle \alpha^2 \rangle
\]
between the mean speckle intensity \( \langle I \rangle \) and the number \( N \) of illuminated particles; \( \alpha \) is determined by the particle properties. Also, Goodman [15, pp. 36–40, Eq. (2.81) on p. 38 in particular] derives the autocorrelation function \( R_I(\Delta x, \Delta y) \) of speckle intensity as
\[
R_I(\Delta x, \Delta y) = \langle I \rangle^2 \left( 1 + \frac{\int_{-\infty}^{\infty} \int_{-\infty}^{\infty} |P(\xi, \eta)|^2 \exp(2\pi i \lambda^{-1} z^{-1}(\xi \Delta x + \eta \Delta y))d\xi d\eta}{\int_{-\infty}^{\infty} \int_{-\infty}^{\infty} |P(\xi, \eta)|^2 d\xi d\eta} \right)^2,
\]
where \( |P(\xi, \eta)|^2 \) is the incident field intensity distribution at the scattering spot, \( \lambda \) is the incident field wavelength, \( z \) is the orthogonal distance between the scattering spot and the observation plane, and \( (\Delta x, \Delta y) \) is the relative displacement between points in the observation plane for which the correlation is computed. The \( x \)- and \( y \)-directions in the observation plane are consistent with Fig. 1.

Speckle measurement and interpretation typically involves a two-dimensional image of the speckle. While the acquisition of 2D speckle is certainly not a problem for standard cameras, the subsequent data processing can involve large data sets, especially if the dynamic range of the image is high for better fidelity. It is conceivable that the industrial material identification or characterization by speckle analysis [16–18] would favor fast image acquisition and processing, provided the reconstruction quality is comparable with the 2D case. Also, applications such as wearable sensors for non-invasive glucose concentration measurement [19] could benefit from simplifying the detector, decreasing the memory usage, the required processing capability and the power consumption.

The purpose of this paper is to demonstrate that a single line of near-field speckle can be used to estimate the sample concentration with more than 78% accuracy. Our model problem covers suspensions of biological cells in phosphate saline buffer (refractive index \( n = 1.33411 \)). As we demonstrate, the maximum and mean speckle intensity, and the speckle size, of line speckle samples is non-robust or outright unreliable in the estimation of the particle concentration. To overcome this, we use several techniques ranging from standard linear principal component analysis (PCA), support vector machines (SVM), to neural networks in the form of stacked sparse autoencoders (SSAE) augmented with softmax layers or with support vector machines. In particular, SSAE compute sparse, high-level representations of line speckle signals corresponding to different particle concentrations. We discuss and compare the performance of our proposed approaches.

Section 2. briefly explains how we compute the near-field speckle throughout this paper, and how a standard statistical analysis of line speckle would perform in the context of particle concentration estimation. In section 3. we describe stacked sparse autoencoders and softmax classifiers, and in section 4. we use these autoencoders, augmented with softmax layers or with SVM, to estimate particle concentrations from line samples of speckle. There, we also compare these concentration estimates with estimates found using linear PCA and SVM. Finally, section 5. contains our conclusions and an outlook to further work.

2. Computation and standard analysis of speckle

We have implemented a parallelized FORTRAN speckle generator that sums up the full Mie scattering contributions from all illuminated particles in the sample. We plan to make the speckle...
generator freely available online, and to describe it and its use in detail in a separate publication [20]. The program does not take into account particle-particle interactions. The model allows particles to be immersed in a medium of arbitrary complex refractive index. At the moment, we only consider an ambient medium with a real refractive index, i.e., an ambient medium that is transparent at the operating wavelength. Also, our model currently does not include any additional scattering structures, such as supporting structures, glass slabs, sample holder etc. We are here interested in the fundamental possibility of extracting information from line near-field speckle samples, and wish to investigate this possibility under ideal conditions. For the same reason, the model assumes direct projection of the speckle pattern onto a 1D or 2D sensing array, with no lenses transforming the image or contributing to the distance of image capture. In our simulation, the image sensing screen is embedded in the ambient medium with the particles. The length $Z$ of direct-transmission path of light through sample is the orthogonal distance between the planar 2D image sensing screen and the planar wavefront that first encounters the sample, see Fig. 1. Since the distance between our image sensing screen and a particle in the sample may vary from zero to the value of $Z$, in this case to $1\text{cm}$, i.e., approximately $18800\lambda_0$, the computed speckle is a combination of both near-field speckle components and far-field speckle components propagated in the ambient medium (recall that we do not include particle-particle interaction in the model.) We choose to call this ‘the near-field speckle’ to stress the fact that we do not perform any near-field-to-far-field transformation of the computed speckle. The situation thus differs from the typical case where only far-field speckle is measured or computed, at a screen external to the sample and many wavelengths away from it. Also, we do not take into account the air-sample interface at the plane where the illumination first contacts the sample. We next refer to Fig. 2, which shows the geometrical conventions used to describe scattering from a single illuminated sphere in the model. The vector $\hat{k}_{\text{inc}} = \hat{z}$ points in the direction of propagation of the incident beam, and the vector $\hat{k}_{\text{cea}}$ points in the direction from the center of the scattering particle to a chosen observation point at the image sensing screen. The angle $\varphi$ is the angle between the scattering plane (the plane spanned by the vectors $\hat{k}_{\text{inc}}$ and $\hat{k}_{\text{cea}}$) and the $xz$-plane, and the angle $\theta$ is the angle between the vectors $\hat{k}_{\text{inc}}$ and $\hat{k}_{\text{cea}}$. With our chosen geometry parameters, the angular ranges of light scattering are from 0 deg to 360 deg for the angle $\varphi$ in Fig. 2 and from 0 deg to arbitrarily close to 90 deg for the angle $\theta$ in Fig. 2. The case $\theta = 0$ is the direct transmission direction, and the reason $\theta$ may be arbitrarily close to 90 deg is that particles in the sample may be arbitrarily close to the image sensing screen. The Mie scattering regime usually occurs for particle diameters above 1/10 of the operating wavelength. The diameter of biological cells ranges essentially from 150nm for small bacteria to 800$\mu$m for large Amoeba cells [21], and atmospheric aerosol particles vary in diameter from a few nanometers to approximately 100$\mu$m [5, Ch. 7]. Thus, our speckle generator is relevant for suspensions of biological cells and for atmospheric aerosols with diameter greater than approximately 55nm, assuming all scattering particles are well-approximated by a single type of homogeneous sphere. Finally, each realization of a sample in our program results in a new spatial distribution of the particles, since these are uniformly randomly distributed throughout the illuminated cylinder. In particular, agglomeration of particles is highly unlikely in our samples.

In the current implementation, we use FORTRAN in combination with OpenMP for the parallelization of the code. To obtain a robust and reliable implementation of the Bessel functions $J_{\nu+1/2}(\kappa)$ and $Y_{\nu+1/2}(\kappa)$, over a wide range of values for integer $\nu$ and double $\kappa$, we have chosen to use the FORTRAN90 version of the SPECFUN library (originally from NetLib) implemented by J. Burkardt [22]. In particular, we use the two functions rjbesl() and rybesl(). We have validated the implementation over ranges of $\kappa$ and $\nu$ that can be expected to occur in practice, by comparing our results to results from both Mathematica and MATLAB. To compute the results presented in this paper, we have run our speckle generator on a machine with the following specifications: $2 \times$ Intel Xeon CPU E5-2650 v4 @ 2.20GHz, 256 GB RAM, Scientific
Linux 7.3, Intel FORTRAN compiler 18.0.3. We use the following parameter values in all our 2D speckle computations here:

- one of 14 particle concentrations $c$: 0.01; 0.10; 0.50; 1.00; 1.50; 2.00; 2.50; 3.00; 3.50; 4.00; 4.50; 5.00; 5.50 or 6.00 million particles per ml. The classes 1–14 referenced in this paper correspond to the above 14 different particle concentrations, in order.
- particle radius 7.5 $\mu$m (biological cell)
- particle refractive index 1.37 (biological cell)
- ambient medium refractive index $n_m = 1.33411$ (phosphate saline buffer)
- operating free-space wavelength $\lambda_0 = 532$nm (green laser)
- length of direct-transmission path of light through sample $Z = 1$cm (see Fig. 1)
- laser beam radius $r_{laser} = 1$mm
- near-field screen side length 6mm
- number of pixels in near-field screen: 2449×2449, i.e., a nearly 6 megapixel camera with 2.45-micron pixels

With these parameters, the number of illuminated particles in the sample varied from approximately 300 to approximately 190000, requiring computation times for 2D near-field speckle from 200 sec to 32 hours.

We shall here validate our Mie speckle generator by comparing the computed 2D mean speckle intensity and 2D speckle size with the theoretical predictions from (1) and (2), respectively. Figure 3(a) shows an example of 2D near-field speckle intensity $I(x, y) = |E_x(x, y)|^2 + |E_y(x, y)|^2$ obtained using our speckle generator. Here $E_x$ and $E_y$ are the $x$- and $y$-components of the total (incident plus scattered) electric field intensity, and $(x, y)$ are coordinates on the image sensing screen, consistent with the axes in Fig. 1. To be able to show a more detailed image of the computed speckle in Fig. 3(a), we omit the pixels in the image sensing screen that register speckle intensities above 0.01. These relatively few pixels are located in the directly illuminated disk portion of the speckle image, which is never included in any of our subsequent analysis.
Figure 4a shows log-transformed speckle intensity distributions (probability densities) for 2D speckle samples at the 14 considered particle concentrations. The probability densities seem to have the same shape, and only the mean of the speckle intensity changes (increases) with increasing particle concentration. Figure 5 shows the mean intensity $\langle I \rangle$, the maximum intensity and the size of the computed 2D near-field speckle, for the 14 considered particle concentrations.

In the 2D case, for Fig. 5(c), we compute the speckle size as follows: we first sample $I_{200 \times 200}$, which is the first 200 rows and first 200 columns of the speckle image, see Fig. 3(a). We then compute the autocorrelation function $A_{200 \times 200}$ of $I_{200 \times 200}$, and finally evaluate

$$\text{speckle size (2D case)} = \sum_{i=1}^{200} \sum_{j=1}^{200} A_{200 \times 200}(i, j).$$

We define speckle size by (3) because the graphs of our computed autocorrelation functions consist of one narrow, high central peak with positive values, surrounded by a large 'flat area' of small fluctuating values. The sum of the values of autocorrelation functions yields, in our case, a good and consistent approximation of the volume under the prominent central peak. Finally, we found it sufficient to base our computation of the mean speckle intensity on the whole 2D image without the directly illuminated region.

The computed mean speckle intensity in the 2D case, Fig. 5(a), is rather convincingly a linear function of the particle concentration, consistent with the prediction in (1), with the slope $V(\alpha^2) = r_{\text{laser}}^2 \pi Z(\alpha^2) \approx 5.83 \cdot 10^{-12}$, corresponding to the particle-dependent factor $\langle \alpha^2 \rangle \approx 1.86 \cdot 10^{-4}$. Next, we compared the computed autocorrelation matrices $A_{200 \times 200}$, for all 14 concentrations, with the predictions from Eq. (2). For Eq. (2), we chose the parameter values $\lambda = \lambda_0/\eta_m = 532\text{nm}/1.33411 \approx 398.77\text{nm}$ and $z = 1\text{cm}$, as well as the incident field intensity $I_{laser}$.

The case $c = 3.5 \cdot 10^6 \text{ml}^{-1}$ is representative, and Fig. 6 shows for this case the log-transformed relative difference between the values $\langle A_{200 \times 200}(101 : 200, 101) \rangle$ and the values from Eq. (2) with $\Delta x \in [0, 6\text{mm}]$, $\Delta y = 0$. This comparison thus involves the values of the computed and the predicted autocorrelation function starting at the top of the prominent central peak ($A_{200 \times 200}(101, 101)$ and $\Delta x = \Delta y = 0$) and along increasing $\Delta x$ displacement, with $\Delta y$ always zero, up to the displacement $\Delta x = 6\text{mm}$. We can allow to restrict the comparison to the above line due to the radial symmetry of our autocorrelation function. All 14 obtained relative differences between the computed and the theoretical autocorrelation function were under $10^{-1.5}$, i.e., under 3.2%.

Figure 3(b) shows the top horizontal line of the speckle image in Fig. 3(a). We call such line samples of speckle images '1D speckle samples.' The densities for 1D speckle samples (b) show clear evidence of being based on significantly fewer speckle intensity values compared to the densities for 2D speckle samples (a). Figure 7 shows the mean intensity $\langle I \rangle$, the maximum speckle intensity and the speckle size for 1D line samples of the computed speckle, for the 14 considered particle concentrations. For Fig. 7, every 1D line sample was taken to be the top horizontal line of the corresponding computed 2D speckle image. In the 1D case, for Fig. 7(c), we compute the autocorrelation function $A_{2449}$ of the full 1D, 2449-pixel signal and define the speckle size to be

$$\text{speckle size (1D case)} = \sum_{i=1}^{2449} A_{2449}(i).$$

We justify our definition (4) of size of 1D speckle using the same argument we presented for the definition (3) of 2D speckle size.
Fig. 3. (a): the logarithm of the computed near-field speckle intensity with particle concentration $c = 6 \cdot 10^6$ ml$^{-1}$, radius $R = 7.5$ µm and refractive index $n = 1.37$. The central bright disk is the region of direct transmission of incident laser light. The 200-by-200 pixel sample $I_{200 \times 200}$ in the top left corner is used in the estimation of 2D speckle mean intensity and 2D speckle size. (b): values from the first horizontal line of the graph in (a).
Fig. 4. Probability densities for computed near-field speckle intensity at 14 particle concentrations. (a): the 2D speckle case. (b): the case with line samples of computed 2D speckle patterns. The directly illuminated region of the image sensing screen is ignored in all cases.
Fig. 5. Statistics for computed 2D near-field speckle patterns as in Fig. 3(a). The directly illuminated portion of the image sensing screen is ignored. (a): mean speckle intensity, as function of particle concentration. (b): maximum speckle intensity as function of particle concentration. (c): speckle size (see Eq. (3)) as function of particle concentration.
Before we begin assessing how well-suited the speckle information in Figs. 5 and 7 is for estimating the particle concentration, let us investigate numerically the effect of different spatial distribution of particles in different realizations of samples with the same concentration. We have computed the speckle intensity distributions, mean and maximum speckle intensities, as well as speckle sizes, for 100 realizations of samples with particle concentration $10^4\text{ml}^{-1}$ and for 100 realizations of samples with particle concentration $10^6\text{ml}^{-1}$. Figures 8 and 9 show the speckle intensity distributions in the two cases. Based on these results, we do not expect significant change in the speckle parameters of Figs. 5 and 7 as the spatial distribution of the particles changes to another instance of uniformly distributed particles. In the case $c = 10^4\text{ml}^{-1}$ and 2D speckle computation, the 100 realizations gave a mean speckle intensity with mean value $5.88 \cdot 10^{-8}$ and variance $1.69 \cdot 10^{-18}$, maximum speckle intensity with mean value $1.04 \cdot 10^{-7}$ and variance $3.44 \cdot 10^{-17}$, and speckle size with mean value $1.39 \cdot 10^{-10}$ and variance $3.75 \cdot 10^{-23}$. In the case $c = 10^6\text{ml}^{-1}$ and 1D speckle, the 100 realizations gave a mean speckle intensity with mean value $5.74 \cdot 10^{-8}$ and variance $2.45 \cdot 10^{-18}$, maximum speckle intensity with mean value $5.43 \cdot 10^{-8}$ and variance $1.09 \cdot 10^{-16}$, and speckle size with mean value $1.06$ and variance $0.01$. In the case $c = 10^6\text{ml}^{-1}$ and 2D speckle computation, the 100 realizations gave a mean speckle intensity with mean value $5.90 \cdot 10^{-6}$ and variance $1.03 \cdot 10^{-15}$, maximum speckle intensity with mean value $1.04 \cdot 10^{-5}$ and variance $4.16 \cdot 10^{-13}$, and speckle size with mean value $1.39 \cdot 10^{-6}$ and variance $2.30 \cdot 10^{-16}$. In the case $c = 10^6\text{ml}^{-1}$ and 1D speckle, the 100 realizations gave a mean speckle intensity with mean value $5.80 \cdot 10^{-6}$ and variance $1.84 \cdot 10^{-14}$, maximum speckle intensity with mean value $5.62 \cdot 10^{-6}$ and variance $1.30 \cdot 10^{-12}$, and speckle size with mean value $1.08$ and variance $0.01$. Our analysis in the rest of this paper is based on a single realization of a 2D speckle pattern for each considered particle concentration.

Figure 5 shows that the mean and maximum speckle intensity and the speckle size of the 2D speckle image can in principle be used to estimate a wide range of particle concentrations. Intuitively, both the maximum and the mean speckle intensity should exhibit an increase with increasing particle concentration, at least over a considerable range of concentrations, since more scattering spheres means more of the incident beam energy being directed away from the directly illuminated portion of the image sensing screen. In the case of 1D speckle sampling, the mean speckle intensity, Fig. 7(a), can in principle be used to estimate the particle concentration. However, in both the 1D and the 2D case, and over the whole concentration range, the variation of
Fig. 7. Statistics for computed single lines of near-field speckle as in the first horizontal line in Fig. 3a, also shown in Fig. 3b. The directly illuminated portion of the image sensing screen is never sampled here. (a): mean speckle intensity, as function of particle concentration. (b): maximum speckle intensity as function of particle concentration. (c): speckle size (see Eq. (4)) as function of particle concentration.
Fig. 8. Probability densities for speckle intensity produced by 100 realizations of particle samples with concentration $c = 10^4 \text{ml}^{-1}$.
Fig. 9. Probability densities for speckle intensity produced by 100 realizations of particle samples with concentration $c = 10^6 \text{ml}^{-1}$. 
the mean speckle intensity is only approximately $3.5 \cdot 10^{-5}$ of the incident field amplitude, which may imply instability in the estimation of concentration and the need for sensitive equipment and noise suppression. The maximum speckle intensity and speckle size, Fig. 7(b) and Fig. 7(c), are manifestly a poor basis for the estimation of the particle concentration $c$. This is perhaps unsurprising; the ratio of the directly illuminated area to the full near-field screen area is $r^2_\text{laser} \pi/a^2 = (1 \cdot 10^{-3})^2 \cdot \pi/(6 \cdot 10^{-3})^2 = \pi/36 \approx 0.087$, so the 1D speckle to 2D speckle data ratio is here approximately only $2449 / [(1 - 0.087) \cdot 2449^2] \approx 4.5 \cdot 10^{-4}$. Figure 4 helps visualize the difference in the data content of 1D and 2D speckle samples. The maximum speckle intensity there appears to have an approximately linear dependence on the particle concentration up to approximately 2 million particles per ml. However, this simple dependence breaks down for higher particle concentrations and, even worse, the maximum speckle intensity attains approximately the same value for several different particle concentrations. This may be since the drastically reduced number of sampled pixels on the image sensing screen (in the 1D sampling case) significantly reduces the chance that the full actual range of the pixel intensities is seen in a single 1D sample. Figure 7c indicates that the speckle size, computed for 1D speckle samples, lacks a systematic dependence on the particle concentration over the whole considered range of concentrations. This may be because, unlike the maximum speckle intensity, the speckle size does not depend on single pixel values, but is rather determined by the correlation of several neighboring pixel values. Reducing the speckle sampling from a 2D image to a single line may destroy a significant portion of the information carried by the 2D image in relation to actual speckle size.

Instead of considering the statistics in Fig. 7, another standard way to classify 1D speckle data would be via linear principal component analysis (PCA) [23,24], a widely used and generally applicable method of dimension reduction for large data sets. We start by collecting a training data set, taking a number of line samples from the computed 2D speckle corresponding to each of the 14 concentrations. Next, we find the principal components by computing the eigenvalues and the eigenvectors of the covariance matrix of the training data set. Figure 10 shows the projection of the training set samples along the first two principal components. Evidently, if we work directly with the computed values of the speckle intensity then the projections of the training data yield no clear classification of any but the lowest of the considered concentrations.

Fig. 10. Projection of training set samples along the first two principal components, (a) prior to and (b) after taking the logarithm of the speckle intensity.
Therefore, we log-transform the speckle intensity to increase the linearity in the data and estimate again the eigenvectors and eigenvalues of the covariance matrix, Fig. 11. The significant spectral gap between the first and the second eigenvalue indicates that the variance of the data is mostly preserved in the direction of the first eigenvector, Fig. 11. Computing the ratio \[ \kappa = \frac{\sum_{i=1}^{N_{PCA}} \lambda_{PCA,i}}{\sum_{i=1}^{D_{PCA}} \lambda_{PCA,i}}, \] we can see that this single direction preserves over 99.9\% of the variance of the data. Here, \( N_{PCA} \) is the number of eigenvalues summed up in the numerator in (5), and \( D_{PCA} \) is the total number of eigenvalues. Thus, the maximum classification success rate for the linear PCA can in this case be obtained using only the first two eigenvectors. We process the projected training data set with a softmax classifier, described in the next section. This results in a supervised learning approach. The performance of this approach is shown in Fig. 14 for original speckle intensity test data, and in Fig. 15 for log-transformed speckle intensity test data. The training data set and the test data set are disjoint. The resulting success rates in concentration classification are 64.10\% and 70.90\%, respectively. Replacing a softmax classifier with a support vector machine [26–28], we can improve the success rate for the linear PCA with log-transformed data to 77.76\%, see Fig. 17. A support vector machine (SVM) is a standard supervised classifier defined by a separating hypersurface. For a given set of labeled training data, this method produces an optimal hypersurface that categorizes samples. Using SVM directly on the log-transformed 1D speckle data produces the success rate of 75.24\%, see Fig. 16.

3. Stacked sparse autoencoder for classification of concentration

A stacked sparse autoencoder (SSAE) [29–32] is a neural network with multiple layers of sparse autoencoders, where the output of each layer is connected to the input of the next layer. Application of an SSAE can be viewed as nonlinear principal component analysis of the input. An SSAE has the architecture as depicted in Fig. 12, and it is trained in an unsupervised manner.
to recreate input data. It consists of three parts: encoder, decoder and latent space/bottleneck. An encoder is a system of connected hidden layers with the purpose to compress the input data to a latent space from which, by a decoder, the input data are again recreated. With reference to Fig. 12, we have

\[ h = g_1(Wx + b) \quad \text{and} \quad \hat{x} = g_2(\hat{W}h + \hat{b}), \]  

(6)

where \( g_1 \) and \( g_2 \) are activation functions, \( W \) and \( \hat{W} \) are weight matrices, and \( b, \hat{b} \) are bias vectors for the encoder and decoder, respectively \[29\].

![Fig. 12. Neural network architecture of a stacked sparse autoencoder (SSAE).](image)

To best recreate the input data, an SSAE is trained to minimize an adjusted mean square loss function \( L \) of the form

\[ L(W, b, \hat{W}, \hat{b}) = \frac{1}{N} \sum_{i=1}^{N} \sum_{k=1}^{K} (x_{ki} - \hat{x}_{ki})^2 + \lambda \cdot (L^2 \text{ overfitting term}) + \beta \cdot (\text{sparsity term}). \]  

(7)

Here \( x \) is the input vector, \( \hat{x} \) the reconstructed vector, \( N \) is the number of training samples, \( K \) is the number of features within a sample, \( \lambda \) is the coefficient of a regularization term that prevents overfitting, and \( \beta \) is the coefficient of a regularization term that promotes sparsity \[29\].

For this work, an SSAE was assembled with five hidden layers consisting of 1200, 400, 100, 50 and 20 hidden units, respectively (Fig. 12). For all layers, the logistic sigmoid function was used for the encoders and the decoders. The layers 1–3 were trained within the maximum of 400 epochs, and with the parameter values \( \lambda = 0.004, \beta = 4 \) and sparsity proportion of 0.15. The layers 4–5 were trained within the maximum of 1000 epochs, and with the parameter values \( \lambda = 0.002, \beta = 4 \) and sparsity proportion 0.1. The network was trained using the scaled conjugate gradient descent function with greedy layer-wise training \[29\]. After the SSAE training, the latent space representation of the input, present in the bottleneck of the SSAE, Fig. 18, was used to train a softmax classifier \[30,31\] over 1000 epochs in a supervised fashion. Prior to the final classification, the encoder of the SSAE and the softmax layer, see Fig. 13, were additionally fine-tuned with extra training epochs. Unlike a support vector machine, whose outputs are classification scores for each class, a softmax classifier gives a probabilistic interpretation \[30,31\]. This is accomplished by using a logistic vector function \( f_i \), called the softmax function, Eq. (8), which takes a vector \( h_i \) of real-valued scores for each of the classes \( i = 1, \ldots, M \), and compresses
it to a vector of values

$$f_i(h) = \frac{\exp(h_i)}{\sum_j^M \exp(h_j)}$$  

(8)

between 0 and 1, the components of which sum up to 1.

**Fig. 13.** The encoder part of a stacked sparse autoencoder (SSAE) with the bottleneck attached to the softmax classifier.

### 4. Numerical inversion of 1D speckle samples

The training data were comprised of 200 1D line samples of near-field speckle patterns, see Fig. 3(b), from each of the 14 computed near-field speckle images corresponding to the 14 considered particle concentrations. The test data were comprised of 150 1D speckle samples from each of the 14 speckle images. For each concentration, we computed and sampled only one 2D speckle image, and such that the training data set and the test data set were disjoint. We wrote the principal component analysis code in MATLAB, and used MATLAB’s Deep Learning toolbox for the training of stacked sparse autoencoders, softmax layers and support vector machines.

Our first approach to concentration classification was to feed the PCA projections, Fig. 10, to a softmax classifier. As already mentioned, PCA ideally exploits any linearity in the data to project the data to a low-dimensional subspace and to expose clustering of the projections. Figure 10 demonstrates an advantage of the log-transformation of our 1D speckle intensity samples, in that the data are somewhat linearized and the projections become more clustered. We fed the projected data to a softmax classifier layer trained over 1000 epochs. We validated the softmax classifier over a set of test data, resulting in the confusion matrices of Figs. 14 and 15. The vertical axis of a confusion matrix describes the output of a trained network classifier, while the horizontal axis gives the true classification of the sample. The ‘accuracy’ value above our confusion matrices is the mean classification performance for a trained network. The diagonal values in a confusion matrix represent the individual classification accuracies for each class. Off-diagonal values indicate false predictions. In the test data, we had 150 samples for each class. The classification accuracy of the application of the PCA-softmax layer on the test data without the log-transformation was 64.1%, and 70.9% with the 1D speckle intensities log-transformed. The classifier without the log-transformation could not distinguish the classes 1 and 2. In all cases, there was difficulty in resolving neighboring classes.
By training an SSAE with a softmax layer, we improved the classification performance to 77.62% on average. While our improvement is a modest 6% on average, it does significantly, by 11.37% on average, boost the classification performance for the classes 9–14, that is, for the high particle concentrations from 3.5 to 6 million per ml. This boost is evident from the confusion matrices in Figs. 15 and 19. It is evident from Fig. 10 that the linear PCA approach has difficulty discerning these high-concentration classes, as expected since the relative difference in concentrations of neighboring classes becomes smaller for increasing concentration. The SSAE encoded and compressed the training data to a 20-dimensional latent space, Fig. 12. This latent space representation, as well as the classification performance of the SSAE, can be visualized using the t-Distributed Stochastic Neighbor Embedding (t-SNE), see Fig. 18. Contrary to principle component analysis, t-SNE is based on a probabilistic interpretation. As explained in the original paper [33], t-SNE minimizes the divergence between two distributions: a distribution that measures pairwise similarities of the input objects and a distribution that measures pairwise similarities of the corresponding low-dimensional points in the embedding.

The performance of our trained SSAE with a softmax layer is shown in Fig. 19. The network classified the cases 1–6, i.e., the concentrations from $c = 10^4$ ml$^{-1}$ to $c = 2 \cdot 10^6$ ml$^{-1}$, as well as class 14, with $c = 6 \cdot 10^6$ ml$^{-1}$, with the average success rate of 95.4%. The network identified the concentrations between $2.5 \cdot 10^6$ ml$^{-1}$ and $5.5 \cdot 10^6$ ml$^{-1}$ with the average success rate of 60.1%. The misclassifications were generally made between neighboring classes. The class 13,
with \( c = 5.5 \cdot 10^6 \text{ ml}^{-1} \), had a success rate of less than 50\%. The network misidentified the class 13 as the class 14 in 38.9\% of instances, and it misidentified the class 8 for class 9 in 42.3\% of instances. The data for these classes were too similar for the network to have better predictions.

To test the classification method in a situation with increased distance between the neighboring classes, we next considered only the 6 classes with concentrations 1, 2, 3, 4, 5 and 6 million particles per ml. The classification accuracy increased to 89.33\%, as shown in Fig. 21.

Replacing a softmax classifier with a support vector machine (SVM) with linear kernel function improves the classification success rates. For the linear PCA with log-transformed data, the classification rate improves to 77.76\%, which is slightly better than the SSAE with a softmax layer, see Fig. 17 and Fig. 19. Comparing with the PCA approach with a softmax layer, see Fig. 15, the improvements for the high-concentration classes 9–14 (3.5 to 6 million particles per ml) are significant and 11\% on average. Combining SSAE with SVM improves the classification rate for 1D speckle intensity to 78.76\%. In particular, this rate also improves for the classes 9–13, see Fig. 20. Before feeding it to the SVM, we log-transform the encoded training data. Even then, class 13 is recognized at a rate of less than 50\%, while for the rest of the classes the classification rates are above approximately 60\%, with 7 of the classes above 75\%. Our results indicate that combining SSAE or PCA with SVM allows better classification of 1D speckle samples than combining the two techniques with a softmax layer. Also, we find that a single line of speckle contains enough information to allow reasonably reliable classification of particle concentration.

**Fig. 15.** Confusion matrix for linear PCA analysis of log-transformed speckle intensity data with a softmax classifier.
Fig. 16. Confusion matrix for support vector machine analysis of log-transformed speckle intensity data
### Fig. 17. Confusion matrix for linear PCA analysis of log-transformed speckle intensity data with a support vector machine (SVM).

<table>
<thead>
<tr>
<th>Target Class</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
<th>13</th>
<th>14</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>100.0%</td>
<td>0.0%</td>
<td>0.0%</td>
<td>0.0%</td>
<td>0.0%</td>
<td>0.0%</td>
<td>0.0%</td>
<td>0.0%</td>
<td>0.0%</td>
<td>0.0%</td>
<td>0.0%</td>
<td>0.0%</td>
<td>0.0%</td>
<td>0.0%</td>
</tr>
<tr>
<td>2</td>
<td>0.0%</td>
<td>100.0%</td>
<td>0.0%</td>
<td>0.0%</td>
<td>0.0%</td>
<td>0.0%</td>
<td>0.0%</td>
<td>0.0%</td>
<td>0.0%</td>
<td>0.0%</td>
<td>0.0%</td>
<td>0.0%</td>
<td>0.0%</td>
<td>0.0%</td>
</tr>
<tr>
<td>3</td>
<td>0.0%</td>
<td>0.0%</td>
<td>100.0%</td>
<td>0.0%</td>
<td>0.0%</td>
<td>0.0%</td>
<td>0.0%</td>
<td>0.0%</td>
<td>0.0%</td>
<td>0.0%</td>
<td>0.0%</td>
<td>0.0%</td>
<td>0.0%</td>
<td>0.0%</td>
</tr>
<tr>
<td>4</td>
<td>0.0%</td>
<td>0.0%</td>
<td>0.0%</td>
<td>100.0%</td>
<td>0.0%</td>
<td>0.0%</td>
<td>0.0%</td>
<td>0.0%</td>
<td>0.0%</td>
<td>0.0%</td>
<td>0.0%</td>
<td>0.0%</td>
<td>0.0%</td>
<td>0.0%</td>
</tr>
<tr>
<td>5</td>
<td>0.0%</td>
<td>0.0%</td>
<td>0.0%</td>
<td>0.0%</td>
<td>97.4%</td>
<td>0.0%</td>
<td>0.0%</td>
<td>0.0%</td>
<td>0.0%</td>
<td>0.0%</td>
<td>0.0%</td>
<td>0.0%</td>
<td>0.0%</td>
<td>0.0%</td>
</tr>
<tr>
<td>6</td>
<td>0.0%</td>
<td>0.0%</td>
<td>0.0%</td>
<td>0.0%</td>
<td>0.0%</td>
<td>2.6%</td>
<td>89.8%</td>
<td>0.6%</td>
<td>0.0%</td>
<td>0.0%</td>
<td>0.0%</td>
<td>0.0%</td>
<td>0.0%</td>
<td>0.0%</td>
</tr>
<tr>
<td>7</td>
<td>0.0%</td>
<td>0.0%</td>
<td>0.0%</td>
<td>0.0%</td>
<td>0.0%</td>
<td>14.2%</td>
<td>72.4%</td>
<td>2.0%</td>
<td>0.0%</td>
<td>0.0%</td>
<td>0.0%</td>
<td>0.0%</td>
<td>0.0%</td>
<td>0.0%</td>
</tr>
<tr>
<td>8</td>
<td>0.0%</td>
<td>0.0%</td>
<td>0.0%</td>
<td>0.0%</td>
<td>0.0%</td>
<td>0.0%</td>
<td>27.1%</td>
<td>60.7%</td>
<td>9.5%</td>
<td>0.0%</td>
<td>0.0%</td>
<td>0.0%</td>
<td>0.0%</td>
<td>0.0%</td>
</tr>
<tr>
<td>9</td>
<td>0.0%</td>
<td>0.0%</td>
<td>0.0%</td>
<td>0.0%</td>
<td>0.0%</td>
<td>0.0%</td>
<td>0.0%</td>
<td>37.3%</td>
<td>67.2%</td>
<td>1.2%</td>
<td>0.0%</td>
<td>0.0%</td>
<td>0.0%</td>
<td>0.0%</td>
</tr>
<tr>
<td>10</td>
<td>0.0%</td>
<td>0.0%</td>
<td>0.0%</td>
<td>0.0%</td>
<td>0.0%</td>
<td>0.0%</td>
<td>0.0%</td>
<td>0.0%</td>
<td>23.4%</td>
<td>60.8%</td>
<td>11.7%</td>
<td>0.0%</td>
<td>0.0%</td>
<td>0.0%</td>
</tr>
<tr>
<td>11</td>
<td>0.0%</td>
<td>0.0%</td>
<td>0.0%</td>
<td>0.0%</td>
<td>0.0%</td>
<td>0.0%</td>
<td>0.0%</td>
<td>0.0%</td>
<td>0.0%</td>
<td>38.0%</td>
<td>57.9%</td>
<td>1.8%</td>
<td>0.0%</td>
<td>0.0%</td>
</tr>
<tr>
<td>12</td>
<td>0.0%</td>
<td>0.0%</td>
<td>0.0%</td>
<td>0.0%</td>
<td>0.0%</td>
<td>0.0%</td>
<td>0.0%</td>
<td>0.0%</td>
<td>0.0%</td>
<td>0.0%</td>
<td>63%</td>
<td>84%</td>
<td>3%</td>
<td>0%</td>
</tr>
<tr>
<td>13</td>
<td>0.0%</td>
<td>0.0%</td>
<td>0.0%</td>
<td>0.0%</td>
<td>0.0%</td>
<td>0.0%</td>
<td>0.0%</td>
<td>0.0%</td>
<td>0.0%</td>
<td>0.0%</td>
<td>0.0%</td>
<td>39.2%</td>
<td>40.5%</td>
<td>15.7%</td>
</tr>
<tr>
<td>14</td>
<td>0.0%</td>
<td>0.0%</td>
<td>0.0%</td>
<td>0.0%</td>
<td>0.0%</td>
<td>0.0%</td>
<td>0.0%</td>
<td>0.0%</td>
<td>0.0%</td>
<td>0.0%</td>
<td>0.0%</td>
<td>0.0%</td>
<td>3.5%</td>
<td>42.6%</td>
</tr>
</tbody>
</table>

**Accuracy:** 77.76%

### Fig. 18. T-Distributed Stochastic Neighbouring Entities (t-SNE) projection of latent space representations for the training set.
**Fig. 19.** Confusion matrix for the SSAE with a softmax layer.
**Accuracy: 78.76%**

<table>
<thead>
<tr>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
<th>13</th>
<th>14</th>
</tr>
</thead>
<tbody>
<tr>
<td>150</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
</tr>
<tr>
<td>0</td>
<td>150</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>100%</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>150</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>150</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>150</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>150</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>150</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>150</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>150</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>150</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
</tr>
</tbody>
</table>

Fig. 20. Confusion matrix for the SSAE with a support vector machine (SVM).
Fig. 21. Confusion matrix for the SSAE with a softmax layer, trained on the concentrations $c = 1, 2, 3, 4, 5$ and $6 \cdot 10^6$ ml$^{-1}$. 

Accuracy: 89.33%
5. Conclusion

Using a sparse stacked autoencoder with a support vector machine, we have achieved a higher than 78% average success rate in the classification, according to particle concentration, of 1D line samples of synthetic near-field speckle patterns for a set of 14 particle concentrations between $10^4$ and $6 \cdot 10^6$ per ml. We achieved slightly lower average success rates using other estimation techniques, including an autoencoder with a softmax layer. Fast, inexpensive, non-invasive and accurate methods for the determination of concentration of particles are needed in various industrial applications. Thus, 1D speckle acquisition with high dynamic range sensors and subsequent analysis deploying a machine learning algorithm may be the approach that combines all these desired attributes. The autoencoder classifier may show improved performance if the size of the training set is increased, and if details of its architecture are improved. For example, one may use a convolutional autoencoder instead of a sparse stacked autoencoder. Finally, examination of the speckle intensities computed by our speckle generator program showed that the pixels in the produced speckle images took on approximately 5.5 million unique values, significantly more than what is available in, say, a 16-bit camera. It will be interesting to see what effect a coarser measurement, both in terms of the number of gray-scale levels and the number of pixels, has on the interpretation of 1D speckle.

Funding

Innovationsfonden (4106-00038B).

References