



OPEN

Comparing machine learning and deep learning regression frameworks for accurate prediction of dielectrophoretic force

Sunday Ajala¹, Harikrishnan Muraleedharan Jalajamony¹, Midhun Nair², Pradeep Marimuthu³ & Renny Edwin Fernandez¹✉

An intelligent sensing framework using Machine Learning (ML) and Deep Learning (DL) architectures to precisely quantify dielectrophoretic force invoked on microparticles in a textile electrode-based DEP sensing device is reported. The prediction accuracy and generalization ability of the framework was validated using experimental results. Images of pearl chain alignment at varying input voltages were used to build deep regression models using modified ML and CNN architectures that can correlate pearl chain alignment patterns of *Saccharomyces cerevisiae* (yeast) cells and polystyrene microbeads to DEP force. Various ML models such as K-Nearest Neighbor, Support Vector Machine, Random Forest, Neural Networks, and Linear Regression along with DL models such as Convolutional Neural Network (CNN) architectures of AlexNet, ResNet-50, MobileNetV2, and GoogLeNet have been analyzed in order to build an effective regression framework to estimate the force induced on yeast cells and microbeads. The efficiencies of the models were evaluated using Mean Absolute Error, Mean Absolute Relative, Mean Squared Error, R-squared, and Root Mean Square Error (RMSE) as evaluation metrics. ResNet-50 with RMSPROP gave the best performance, with a validation RMSE of 0.0918 on yeast cells while AlexNet with ADAM optimizer gave the best performance, with a validation RMSE of 0.1745 on microbeads. This provides a baseline for further studies in the application of deep learning in DEP aided Lab-on-Chip devices.

Tools like DL and ML are integral part of artificial intelligence¹⁻³. ML for image analysis typically involves extraction of important features from an image and training a machine learning model⁴. Machine learning can be highly efficient when the extracted features distinctly represent a particular image. Images need to be converted into feature vectors and train a model⁴⁻⁶. are examples of approaches where machine learning has been used to predict the presence, absence or possibility of an occurrence in images. However, extraction of significant features from complex images is intricate. Alternatively, deep learning does not depend on an input feature. Rather, DL models identifies significant features from processed images and classifies them based on the identified features. Feature maps extracted through deep learning from computed tomography (CT), magnetic resonance imaging (MRI), positron emission tomography (PET), mammography, ultrasound, and histopathology, provide valuable information^{4,7,8}. In cellular biology, DL-based approaches are primarily adopted to detect changes in cell morphology and correlate them to the mechanisms governing drug response^{7,8}. Images of brain, prostate, retina, lungs are often combined with deep learning algorithms to predict medical conditions. U-Net, ResNet, and VGG are the most frequently used Convolutional neural network-derived networks for medical image segmentation and classification tasks. Recently, transfer learning and GAN-derived networks were widely applied in COVID-19 studies. Although, DL training involves intense data processing and long training time, it gives accurate predictions when used with high performance GPU and labelled data. In this study, we have designed models using both machine learning and deep-learning approaches to estimate the magnitude of dielectrophoretic force from microparticle alignment in a point-of-care device.

¹Department of Engineering, Norfolk State University, Norfolk, USA. ²APJ Abdul Kalam Technological University, Thiruvananthapuram, India. ³Rajeev Gandhi College of Engineering and Technology, Puducherry, India. ✉email: refernandez@nsu.edu

Significance. Application of DEP in point-of care sensing devices demands two important requirements— (1) Low voltage (< 10 Voltage) physical device (2) an intelligent system that can correlate microparticle pearl-chain formation into dielectrophoretic force.

The dielectrophoretic force (F_{DEP}) invoked on a microparticle can be directly correlated to its dielectric property changes (Eq. 1). The DEP force is also proportional to the electric field intensity, particle dimension and medium conductivity^{9–12}. Practically, particle alignment with respect to the electrodes, at a particular voltage and frequency is taken as an indicator of DEP force. Although particle alignment differs from experiment to experiment, some of the features of the particle aggregates are dominant and unique. F_{DEP} exerted on microparticles drive them into pearl chain assemblies which eventually will be aligned along the electric field^{13,14}. For instance, the number of particles in a pearl chain at an applied voltage has been found relatively constant. The pattern has been confirmed by several researchers in the past. In an experiment with 5 μm PS beads¹⁵, pearl chains with 10–12 beads were formed for an applied potential of 15 V_{pp} at 200 kHz. Likewise, 10 μm PS beads formed pearl chains with 7–12 beads at 20 V_{pp} at 20 MHz in a low conductivity buffer (1.8×10^{-4} S/m)¹³. In¹⁶ negative DEP of the PS beads was observed when a voltage of 3.8 V_{pp} of 480 kHz frequency was applied, forming 6–7 beads-long pearl chain. Similar studies on yeast cells have been reported when voltage (3.7 V_{pp}) at a field frequency of 580 kHz exhibited positive DEP, the number of particles aggregated were found to be related to the applied voltage^{16,17}.

Most of the studies have been reported on Lab-on Chip devices with precise geometrical features. Electrode aided DEP systems are mainly based on planar electrodes^{18–21} which require voltages between 10–20 V. Electrodeless DEP (i-DEP) systems, operated at voltages in the 20–100 V range, are used for studying the dielectric variations of biological cells and polymer beads^{22,23}. Advances in nanofabrication and nanotechnology have resulted in several DEP aided Lab-on-a-chip (LOC) systems that are used for microparticle separation^{24–26}, identification^{21,27}, concentration²⁸. Several studies have reported dielectric characterization of biological particles such as viruses, bacteria, fungi, protozoa, proteins, lipids, and DNA^{21,29–32} using DEP aided LOC systems. DEP aided immunological sensing is advantageous as they increase the local concentration of target particles thereby enhancing the overall device sensitivity and response time³³. DEP aided LOC has been utilized in conjunction with scanning electron microscopy (SEM) to capture and immobilize viable cells in the SEM region without the use of a chemical surface modification³⁴.

The requirement to precisely estimate F_{DEP} induced on particles is critical in the development of efficient microfluidic systems. F_{DEP} estimation is a necessity in DEP aided dielectric characterization systems used in clinical diagnostics of biological cells³². Patterns of microparticle aggregation is a direct indicator of F_{DEP} and can be correlated to the dielectric properties of microparticles including biological cells^{21,22,31,35,36}. However, due to drag force phenomenon, the yield of LOC devices is extremely low at higher flow rates³⁷. To circumvent this constraint, the F_{DEP} induced on the particles should be increased substantially^{31,32,38} resulting in a need for 3D microelectrodes. The cost of creating LOC systems with 3D electrodes involve nanofabrication techniques that are prohibitively high. We have developed an alternative by using textile electrodes that can invoke high F_{DEP} due to its high surface area.

In our previous work^{12,17}, we have reported the use of textile electrode-based DEP device in conjunction with deep learning to predict the F_{DEP} induced on PS microbeads where we cast the F_{DEP} estimation task as a classification problem. Convolutional Neural Networks (CNNs) classifiers were used to estimate the applied DEP voltage from the pearl chain alignment of the microbeads. This was done to tackle the limitations in the existing F_{DEP} estimation techniques such as the equivalent dipole (EDM)^{39,40}, Maxwell stress tensor (MST)^{41–44}, iterative dipole moment (IDM)^{9,40,45,46} or velocity tracking method. We had used a classification-oriented CNN method in training the CNN models, which though gave excellent training results, performed poorly on testing with adversarial samples^{12,17}. Also, classification paradigm may be inadequate for accurate F_{DEP} computation as the spacing between distinct voltages is crucial. This can only be modeled accurately using a regression framework.

In this work we have treated the F_{DEP} estimation task as a regression problem^{47–49} and have used ML and DL models for analyses. Regression models in computer vision cover a wide range of scenarios, including head-position estimation⁵⁰, face landmark recognition⁵¹, and age estimation^{52,53}. Regression is commonly used to tackle problems requiring the prediction of continuous values. CNN-based deep regression approach has been adopted for cell counting using micrographs^{48,54}. In such situations, the softmax layer is often replaced with a fully connected regression layer with linear or sigmoid activations. Also, the softmax loss which is widely used for classification tasks is replaced with Euclidean loss for regression⁴⁸.

In this research, we have explored ML algorithms like Random Forest, KNN, MLP, SVM, and Linear Regression to determine the F_{DEP} experienced by polystyrene beads and yeast cells in a low conductivity buffer. Because of the ML algorithms especially on out-of-sample and adversarial samples, we further experimented extensively with DL architectures which has the capacity to learn complex data representations with greater accuracy. Using ML approach as a bench mark we explain how deep learning may be used to accurately determine the F_{DEP} . AlexNet, ResNet-50, MobileNetV2, and GoogLeNet are the four pre-trained CNN architectures in this study that were modified into deep regression CNN architectures. We trained the models to predict applied voltages from micrographs of polystyrene and yeast cell pearl chains generated during dielectrophoresis using transfer learning. The abbreviations and words listed in Table 1 will be used throughout the rest of the paper.

Materials and methods

Dielectrophoretic force estimation theory. The interaction of a non-uniform electric field with a dipole is known as dielectrophoretic force F_{DEP} . To improve the approximation of the F_{DEP} exerted on the particles in terms of the voltage applied, a more tangible and straightforward model is required to advance DEP aided sensing systems. The F_{DEP} when the particle is significantly smaller than the non-uniformities in the electric field is given in¹⁷ as:

Abbreviation	Meaning
ML	Machine learning
DL	Deep learning
DEP	Dielectrophoretic
CNN	Convolutional neural networks
PS	Polystyrene
KNN	K-nearest neighbor
SVM	Support vector machine
MAE	Mean absolute error
MRE	Mean absolute relative
MSE	Mean squared error
RMSE	Root mean square error
CT	Computed tomography
MRI	Magnetic resonance imaging
PET	Positron emission tomography
i-DEP	Electrodeless DEP
LOC	Lab-on chip
SEM	Scanning electron microscopy
EDM	Equivalent dipole
MST	Maxwell stress tensor
IDM	Iterative dipole moment
AC	Alternating current
ID	Inner diameter
OD	Outer diameter
ReLU	Rectified linear unit
ADAM	Adaptive moments
RMSProp	Root Mean square propagation
SGDM	Stochastic gradient descent with momentum
AdaGrad	Adaptive gradient
Li	Pearl chain lengths

Table 1. A list of abbreviations and terms used in the paper.

$$F_{\text{DEP}} = 2\pi a^3 \varepsilon_m \text{Re}(f_{cm}) \nabla E_{rms}^2 \quad (1)$$

where a is the radius of spherical microparticles in a medium, under an alternating current (AC) field E_{rms} ; F_{DEP} depends on the product of the localized field with its gradient (∇E_{rms}^2) and the frequency-dependent complex dielectric contrast of the particle versus the medium, as given by real-part of the Clausius–Mossotti factor $\text{Re}(f_{cm})$;

$$f_{cm} = \text{Re} \left(\frac{\varepsilon_p^* - \varepsilon_m^*}{\varepsilon_p^* + 2\varepsilon_m^*} \right) \quad (2)$$

where ε_p^* and ε_m^* are the complex permittivities of the microparticles and the medium, respectively; and given as:

$$\varepsilon_p^* = \varepsilon_p - i \frac{\sigma_p}{\omega} \quad (3)$$

$$\varepsilon_m^* = \varepsilon_m - i \frac{\sigma_m}{\omega} \quad (4)$$

where $i = \sqrt{-1}$ and ω is the angular frequency of the applied AC field.

The DEP fingerprints, or dielectric properties, of a particle in a certain media can be determined by altering the AC signal frequency. Particles can be manipulated once their DEP fingerprints have been discovered. For particle chains, F_{DEP} can be theoretically calculated⁵⁵ using multipole re-expansion and the method of images. F_{DEP} on a particle chain is highly dependent on the angle between the applied field and the chain. The maximum attractive and repulsive forces in a chain grow significantly with the number of particles in the chain, but when the number of particles is large enough, they reach saturation. This F_{DEP} was analytically calculated and given as¹⁷:

$$F_{\text{DEP}} = \varepsilon_E \varepsilon_0 E E_n - \frac{1}{2} \varepsilon_E \varepsilon_0 E^2 n \quad (5)$$

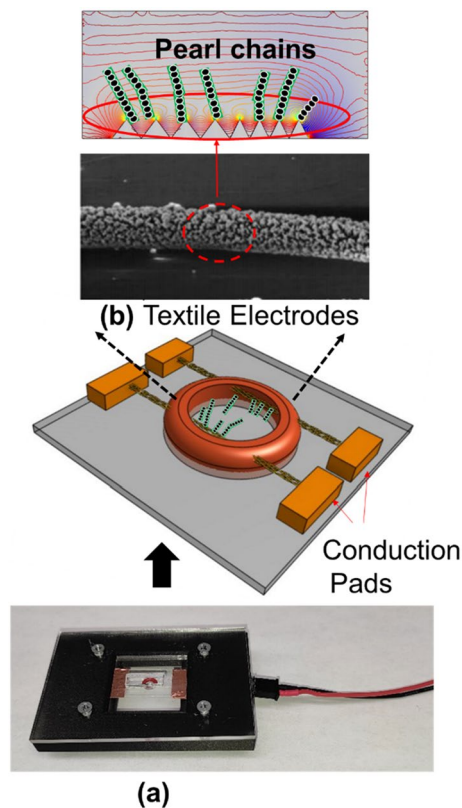


Figure 1. (a) Textile electrode-based DEP device with the connection base (b) SEM micrograph of textile electrodes.

where ϵ_E the relative permittivity of the medium, E is the electric field on the microparticle surface, E_n is the normal component of E , and \mathbf{n} is the unit normal vector on the surface. However, theoretical estimate of F_{DEP} from micrographs is problematic due to discrepancies in the thread structure. At the microscopic level, the orientation of textile strands differs greatly. The calculated force becomes ambiguous as a result.

The applied voltages are predicted by examining the patterns of pearl chain orientation. From the micrographs collected, the deep learning regression algorithms predicted the applied voltage on particles. The force on a chain of spherical dielectric particles in a dielectric fluid is proportional to the number of particles as well as its orientation to the electric field, according to various studies^{55,56}. As a result, a direct link between the applied voltage and the pearl chain formation has been established.

Problem formulation. Let us assume that the j -th image is defined in an input space $x_j \in X$, and there is an output space $y_j \in Y = \{u_1, u_2, \dots, u_k\}$ with sorted ranks $u_k \gg u_{k-1} \gg \dots \gg u_1$. The symbol \gg represents how different rankings are ordered. Given a training dataset $\chi = \{x_i, y_i\}_{i=1}^N$, the goal of regression is to create a mapping from pearl chain images to ranks $g(\cdot) : X \rightarrow Y$ such that the risk functional $R(g)$ is minimized using a specified cost $c : X \times Y \rightarrow R$. The cost matrix C is used to calculate the difference in cost between predicted and ground-truth ranks in this research⁵³. C is a $K \times K$ matrix with $C_{y,u}$ denoting the cost of predicting a sample (x, y) with rank u . Normally,

when $u \neq y$, $C_{y,u} > 0$ and $C_{y,y} = 0$ are assumed. For general regression issues, the absolute cost matrix, which is defined as $C_{y,u} = |y - u|$, is a frequent choice. When applying regression techniques to F_{DEP} estimation, each voltage is treated as a rank.

Machine learning aided pearl chain detection from DEP micrographs. The DEP framework device (Fig. 1) comprises flexible textile electrodes sewn through a silicon O-ring (ID: 1 mm, OD: 3 mm). The textile electrodes are silver-coated conductive string, 82% nylon, and 18% silver. This structure was mounted on a 1×1 inch glass slide. Strings were secured using copper tape, which acted as an electrical contact. Tests were performed by introducing 10 μ L of fluid into the O-ring chamber. A 3D printed custom microscope stage encloses the whole gadget for recording pictures. The pearl chain formations were recorded at different voltages at a fixed frequency of 200 kHz (Fig. 1b). During our dielectrophoresis experiments with yeast cells and 10–20 μ m sized PS microbeads using this setup, 200 micrographs were collected at each voltage level from 1–10 V for yeast cells and polystyrene microbeads, making a sum of 4000 images.

Yeast cells (*Saccharomyces cerevisiae*) are grown in an incubator at 30 °C. The growth medium yeast extract peptone dextrose consisted of 20 g/l peptone, 10 g/l yeast extract, and 20 g/l dextrose dissolved in deionized (DI)

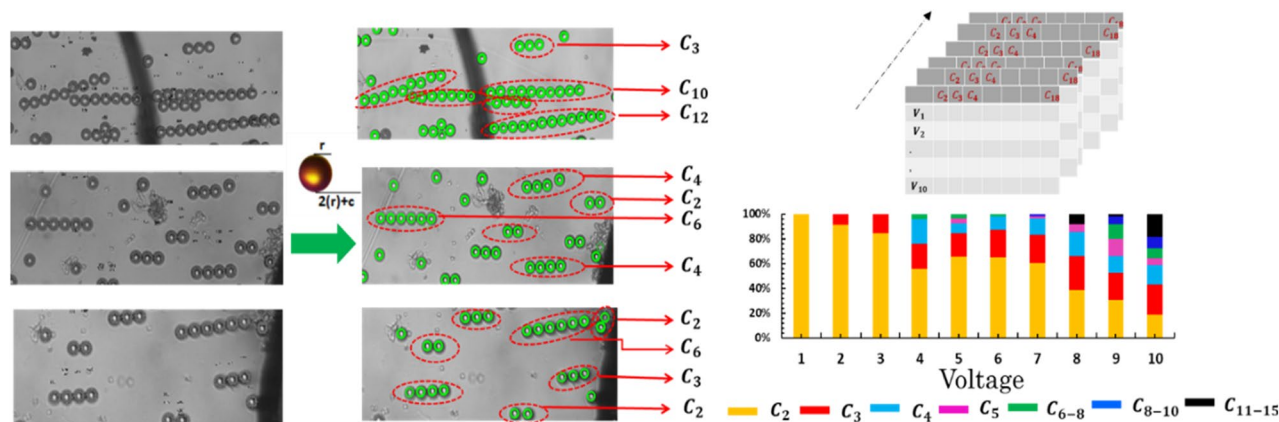


Figure 2. Pearl chain analysis using a Template-Based Matching algorithm and particle coordinate search algorithm. A template image is shifted across the DEP micrographs by an offset (x, y) using the origins of the two images as reference points. Pearl chain lengths (L_i) is determined using a particle coordinate search algorithm.

water. The cells were collected at the stationary growth phase after 1 day of culture in shaking incubator, and they were harvested by centrifugation for 2 min at 3000 rpm and re-suspended in measurement buffers. Plain polystyrene (PS) beads (10, 20 μm) were purchased from Spherotech, Inc., USA. PS beads are charge neutral and are hydrophobic. There was no surface functionalization used. The buffer did not include any surfactant.

Low conductivity buffer: All the microparticles were suspended in an isotonic buffer consisting of 200 mM sucrose, 16 mM glucose, 1 M CaCl_2 , and 5 mM Na_2HPO_4 in DI water (pH 7.4) for the experiments.

Feature extraction for machine learning based regression analysis. We designed a template matching algorithm for object detection using *OpenCV* (algorithm I) to extract the total number of pearl chains in an image, count each pearl chain and map them into a matrix that represents these features. (Fig. 2). Pearl chains are identified within the image using reference shapes which are the cropped images of individual microparticles which is the recognition template. The image dimensions of the template image are also extracted i.e. height, width, to calculate the radius of the microparticle. The radius of a sample pearl in unit of pixels is calculated as $r = (l + b)/4$ where l is the length and b is the breadth of the template image of microparticle. In the formula c is a constant which is fixed at $1/4$ of r . The value of c can be corrected until the output data set includes the data of undetected pearl chains.

The input image is represented as $I(x, y)$, with (x, y) denoting the pixel coordinates. $T(x', y')$ denotes the coordinates of each pixel in the template. Template-Based Matching is done by simply moving the center (or the origin) of the template $T(x', y')$ over each (x, y) point in the input image and calculate the sum of products between the coefficients in $I(x, y)$ and $T(x', y')$, over the whole area spanned by the template. As all possible positions of the template with respect to the input image are searched, the position with the highest score is the best position. In the *OpenCV* implementation, for each location of T over I , we store the cross correlation metric (TM_CCORR_NORMED) in the result matrix R . The cross correlation metric (TM_CCORR_NORMED) used is depicted mathematically as $R(x, y)$ in Eq. 6. Each location (x, y) in R contains the match or cross correlation score, which is the result of sliding the patch with a metric TM_CCORR_NORMED . The brightest locations indicate the highest matches.

$$R(x, y) = \frac{\sum_{x', y'} T(x', y') \cdot I(x + x', y + y')}{\sqrt{\sum_{x', y'} T(x', y')^2 \cdot \sum_{x', y'} I(x + x', y + y')^2}} \quad (6)$$

The method *matchTemplate()* in the *OpenCV* library was used to compare the template image with the input images. The external libraries *cv2*, *numpy*, *glob* and *workbook* were also used. The detected microparticles are marked and corresponding coordinates are stored. Individual pearls are marked in the image using *imwrite()* method. The coordinates obtained are combined with the value of the radius of the pearl, which is then used to identify the pearl chain. After the function finishes the comparison, the best matches can be found as global maximums (TM_CCORR_NORMED) using the *minMaxLoc* function. In case of a color image, template summation in the numerator and each sum in the denominator is done for all the channels. The result will still be a single-channel image, which is easier to analyze. The center coordinate of each microparticle is extracted from the coordinate data set. Each coordinate is used to search for the adjacent microparticle using the condition $2r + c$. All the microparticles nearest to a chain are identified and grouped to a dataset, duplicates are removed, and pearl chains are categorized into pearl chain count, C_L where $L = [2, 3, 4 \dots 18]$. Identified pearl chains are stored in a list, which at the end of processing the pearl chain length and count is stored in an excel sheet. The

precision of image detection can be controlled by changing the values of the threshold in the code. Method `excelWrite()` is used for representing the data of bulk image processed in excel sheet.

Algorithm 1: Pearl Chain Detection

```

Input: image ← input image,
template ← input template image
Output: pearl chain length, count in excel and marked images
1: Procedure processImage:
2: image ← imread input image with cv2
3: template ← imread template with cv2
4: height, width ← shape method of template
5: radius ← (height + width) / 4

6: res ← get list of coordinate using cv2.matchTemplate using TM_CCOEFF_NORMED
7: loc ← filtered res using threshold value with numpy
8: initialize x and y as empty array
9: # Remove noise:
10: for each pt from loc do
11: draw circle on image at (pt[0] + radius, pt[1] + radius) with cv2
12: if pt not duplicate value then
13: x ← append pt[0], y ← append pt[1]
14: end if
15: end
16: # Remove noise:
17: initialize i ← 0
18: while i < length of x do
19: initialize j ← i + 1
20: while j < length of x do
21: if i equals noise then
22: x ← x - x[i]
23: y ← y - y[i]
24: else
25: j ← j + 1
26: end if
27: end
28: i ← i + 1
29: End
30: # Detect chains:
31: set i ← 0
32: initialize chains ← []
33: while i < length of x do
34: x ← x.remove(x[i])
35: y ← y.remove(y[i])
36: chain ← coordinate.axis(x[i],y[j])
37: for ch in chain do
38: set j = 0, c = radius/4
39: while j < length of x do
40: if ch.x < ( x[j] + (2 * radius) + c ) and ch.x > ( x[j] - (2 * radius) - c ) then
41: if ch.y < ( y[j] + (2 * radius) + c ) and ch.y > ( y[j] - (2 * radius) - c ) then
42: chain ← chain.append( Coordinate.Axis(x[j], y[j]) )
43: x ← x.remove(x[i])
44: y ← y.remove(y[i])
45: else
46: j ← j + 1
47: end if
48: else
49: j ← j + 1
50: end if
51: end
52: chains ← chains.append(chain)
53: end
54: initialize chainCount = []
55: # Get the length of chain:
56: for chn in chains do
57: chainCount ← append length of chn
58: end
59: # Count chains with same length:
60: chainCounter = Counter(chainCount)
61: chainCounter → excelSheet
62: imwrite image with cv2 → img file

```

Machine learning models trainings. Prediction of target variable (applied voltage) was done using 18 predictors—which are the pearl chain features. Each predictor represents the number of particles in a pearl chain. Predictors of our model were extracted from the micrographs (Fig. 2). The value of each predictor is the number of microparticles in a pearl chain at an applied voltage. Pearl chain count, C_L where $L = [2, 3, 4 \dots 18]$ represents the number of pearl chains of a specific chain length L , C_L values for all images taken at different voltages are stored in as a matrix and used as features or predictors in order to represent the DEP force. The pearl chain formations were recorded at different voltages. We hypothesize that pearl chain counts C_L in a micrograph at

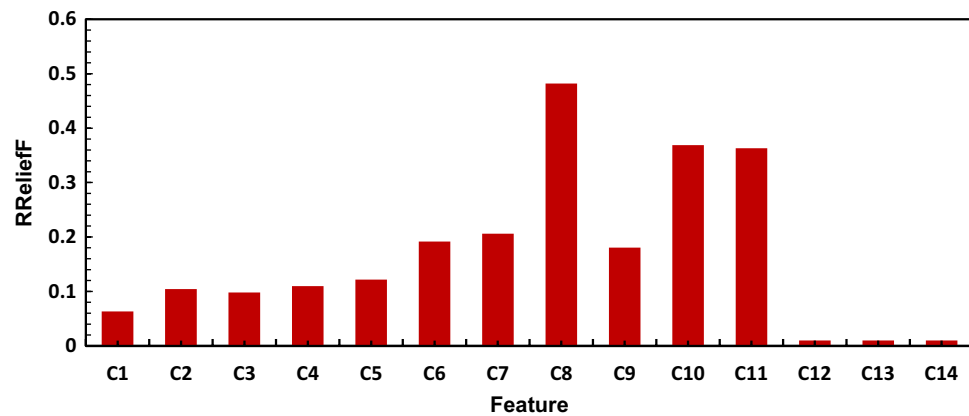


Figure 3. Ranking the importance of the predictors using *RReliefF* algorithm to find a subgroup of high-prevalence features.

different voltages V_x where $1 \leq x \leq 10$, is a thorough representation of DEP micrographs. Evident from the micrographs, pearl chain formations were observed at voltages as low as 2 V. However almost all the pearl chains had not more than 2 microparticles (C_2). At 3 V, 84% of the pearl chains were C_2 and 15% of them were C_3 . For voltages beyond 5 V, $\sim 40\%$ of the pearl chains have more than 4 microparticles (C_4). Above 7 V, $C_8 - C_{10}$ is significant (6.7%). In the 7–10 V range, $C_2 - C_5$ percentages were very low and majority of the pearl chains had more than 8 microparticles C_8 .

ML analyses were performed using the Orange toolbox by writing Python scripts accessing the Orange API. Additional functionalities like feature importance were developed using Python Script widgets. 80% of dataset is assigned as training data set and 20% to testing. Missing values were replaced with the median value of the features. Features with higher dominance in predicting the targets are identified from training sets. As shown in Fig. S1, different ML architectures such as K-Nearest Neighbor (KNN), Support Vector Machine (SVM), Random Forest, Neural Networks, and Linear Regression were trained on the dataset extracted from the PS microbeads micrographs. The Python scripts used for the machine learning is made available via this Github Link (https://github.com/skmidhun09/image_detection_python).

Feature importance estimation for maximum relevance and minimum redundancys. Extraneous features degrade the performance of a model while also increase computing costs. It is critical to find a subgroup of high-prevalence features. Some of the features have a considerable impact on the response model than others. We ranked the importance of features or predictors using *RReliefF* algorithm with k-nearest neighbors. *RReliefF* is a function that works with continuous target. *RReliefF* penalizes features who offer different values to neighbors with the same response values, and rewards features who give different values to neighbors with different response values. Figure 3 shows the features importance ranking obtained by implementing *RReliefF*. Among the C_L predictors where $L = [2, 3, 4, \dots, 18]$, C_8 was found to be the most important feature with a weight of 0.48, followed by C_{10} and C_{11} with importance weight value of 0.37 and 0.36 respectively. C_1 had the least score of 0.06 and $C_{12} - C_{14}$ were found to be insignificant.

Convolutional neural networks as base architecture for deep regression. Convolutional Neural Networks (CNN) was used to extract local trends from spatio-temporal patterns of pearl chain formation. CNNs have at least one layer that uses the convolution operation to extract features^{57,58}. CNNs are used in image processing applications, including automated histopathological image segmentation⁵⁹, automated reconstruction of low-contrast image such as magnetic resonance imaging (MRI)^{60,61}, quantify cyanobacteria from hyperspectral images⁶², medical image processing for direct disease diagnosis^{63,64}, as well as in other disciplines including speech recognition^{58,65} and weather forecast^{57,66}.

We have used four pre-trained CNN architectures viz. AlexNet⁶⁷, MobileNetV2^{68,69}, GoogLeNet⁷⁰ and ResNet-50⁷¹ as the base architectures for deep regression analysis. Table 2 presents a brief overview of these pre-trained CNN architectures. All these architectures were initialized as pre-trained version of the networks which were initially trained on ImageNet dataset for classification. As illustrated in Fig. 4, the pre-trained CNN architectures consist of an input layer, which represents the pixel matrix of an input image, followed by a series of convolution layers that uses Rectified Linear Unit (ReLU) activation. Between two convolution layers is a pooling layer, where max pooling operation is done to down-sample the convoluted image (feature map). Subsequently a fully connected layer where all the inputs are connected along with softmax layers. In order to retrain these pre-trained networks for regression, we remove the last softmax layers from the base architectures (AlexNet, MobileNetV2, GoogLeNet and ResNet-50), employed in the context of classification, and then replace the final fully connected layer, the softmax layer, and the classification output layer with a fully connected layer of size 1 (the number of output variable) with linear activations and a regression layer. As a result, the last layer is a regression layer, whose output dimension corresponds to that of the target space.

Architecture	Main Finding	Number of Parameters	Depth	Numbers of Layers	Input Size	Dataset	Error Rate
AlexNet	Uses Dropout and ReLU	61 million	8	25	227 × 227 × 3	ImageNet	16.4
ResNet-50	Resistant to overfitting, due to symmetry mapping-based skip linkages	25.6 million	50	177	224 × 224 × 3	ImageNet	3.57
MobileNetV2	Inverted residual framework	3.5 million	53	154	224 × 224 × 3	ImageNet	-
GoogLeNet	Block and concatenation concepts, varied filter size, increased depth	7 million	22	144	224 × 224 × 3	ImageNet	6.7

Table 2. Brief overview of the pre-trained CNN architectures.

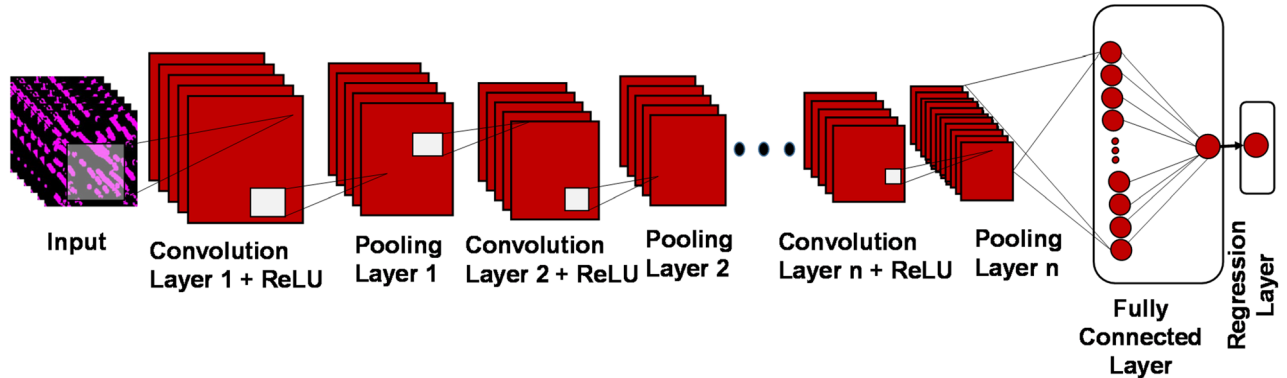


Figure 4. Modification of the Convolutional Neural Networks base architectures (viz: AlexNet, MobileNetV2, GoogLeNet and ResNet-50) from the conventional classification framework to a regression framework.

Hyperparameters	Value
Batch size (n_b)	10
Learning rate (α)	0.0001
Optimizers	RMSProp, SGDM, ADAM
Number of Epochs	100
Validation frequency	30
Execution environment	CPU

Table 3. Hyperparameters used.

Optimizer	Update Rule	Description
RMSProp	$V_{dW} = \beta V_{dW} + (1 - \beta)dW^2$ $W = W - \alpha \frac{dW}{\sqrt{V_{dW} + \epsilon}}$	i. An extension of gradient descent called Adaptive Gradient, or AdaGrad ii. It avoids drastically lowering learning rates by converting the gradient accumulation to an exponentially weighted moving average iii. For that weight, RMSProp only considers recent gradients
ADAM	$S_{dW} = \beta_1 S_{dW} + (1 - \beta_1)dW$ $V_{dW} = \beta_2 S_{dW} + (1 - \beta_2)dW^2$ $Scorr_{dW} = \frac{S_{dW}}{(1 - \beta_1)^T}$ $Vcorr_{dW} = \frac{V_{dW}}{(1 - \beta_2)^t}$ $W = W - \alpha \frac{Scorr_{dW}}{\sqrt{Vcorr_{dW} + \epsilon}}$	i. ADAM is an improvement to the RMSProp optimizer that incorporates momentum method ii. It is an algorithm for handling sparse gradients in noisy problems iii. ADAM is simple to set up, and the default settings work well for most problems
SGDM	$S_{dW} = \beta S_{dW} + (1 - \beta)dW$ $W = W - \alpha S_{dW}$	i. The SGDM approach aids in the acceleration of gradient vectors in the proper directions, resulting in faster convergence ii. It requires more training time and requires hyperparameter tuning than ADAM and RMSProp iii. It is more effective than traditional gradient descent

Table 4. Overview of optimization algorithms.

Notable hyperparameters such as the learning rate α and batch size n_b were appropriately tuned to minimize the cost function and speedup optimization while ensuring the models converge to the global minimum, thereby solving the problem of overfitting^{69,72}. Table 3 presents the CNN hyperparameters used. A series of adaptive learning rate algorithms have recently been developed, Adaptive Moments (Adam)⁷², root mean square propagation (RMSProp)^{72,73}, and stochastic gradient descent with momentum (SGDM)⁷⁴ optimizers explored in this work are among the most widely used optimization algorithms. Table 4 presents a concise overview of these algorithms.

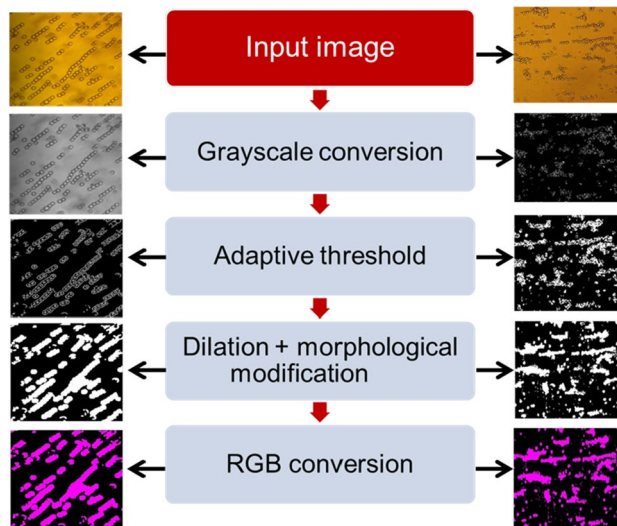


Figure 5. A flowchart of morphological operations carried out on the micrographs for image segmentation, the output segmented image $O(u, v)$ is generated by concatenating $I^\theta(u, v)$ thrice to form the equivalent true color (RGB) image.

The image files collected from micrographs of pearl chain formation at various voltages ranging from 1 to 10 V were used to perform deep learning analysis.

Image preprocessing and segmentation. In order to improve the computational time and accuracy, we applied an optimum adaptive threshold method to reduce the complexity of pearl chain images⁷⁵. Figure 5 depicts the flowchart of the image analysis and segmentation technique. Also, we summarized all of the steps of the segmentation process in Algorithm II.

The MATLAB Image Processing Toolbox was utilized to prototype the methods for image processing in this research. Grayscale conversion, adaptive thresholding segmentation, and morphological operations were all part of the image processing procedures were carried out systematically for 4000 images^{5,76–78}. $I(u, v)$ are grayscale images of the pearl chains in a Euclidean space E . Through intensive thresholding of the pearl chain regions from the micrograph, adaptive global threshold was employed to accomplish segmentation of the pearl chains. The threshold value selected was obtained through image histogram to produce the binarized output $I^\beta(u, v)$.

$$I^\beta(u, v) = \begin{cases} 1 & I(u, v) \geq \alpha \\ 0 & I(u, v) < \alpha \end{cases} \quad (7)$$

α is the adaptive threshold value applied to original input image $I(u, v)$ to get the resultant image, denoted with $I^\beta(u, v)$. In the next steps, a morphological operation called dilation is applied on $I^\beta(u, v)$ using structuring element S_1 (an array of horizontal and vertical lines). The dilation of $I^\beta(u, v)$ by S_1 is mathematically defined as in^{76,77} by Eq. (8) and (9) below, where \hat{S}_1 is the translation of the array S_1 by the vector z and \emptyset is a null set:

$$I^\gamma(u, v) = I^\beta(u, v) \oplus S_1 \quad (8)$$

$$I^\gamma(u, v) = \left\{ z \in E \mid \left(\hat{S}_1 \right)_z \cap I^\beta(u, v) \neq \emptyset \right\} \quad (9)$$

Now, performing a morphological closing operation defined in^{77,78} as the erosion of $I^\gamma(u, v)$ by a horizontal structuring element S_2 , followed by dilation of the resulting image by S_2 , we obtain $I^\delta(u, v)$ as shown below:

$$I^\delta(u, v) = I^\gamma(u, v) \circ S_2 \quad (10)$$

$$I^\delta(u, v) = \left\{ z \in E \mid \left(\hat{S}_2 \right)_z \cap \left(\left\{ z \in E \mid \hat{S}_2 \subseteq I^\gamma(u, v) \right\} \neq \emptyset \right) \right\} \quad (11)$$

Then, we fill the holes of the pearl chains and cleared its border. Then taking a morphological closing operation defined in⁷⁷ as the dilation of $I^\delta(u, v)$ by a horizontal structuring element S_2 , followed by erosion of the resulting image by S_2 , we obtain as shown below:

$$I^\theta(u, v) = I^\delta(u, v) \cdot S_2 \quad (12)$$

$$I^\theta(u, v) = \left\{ z \in E | \widehat{S}_2 \subseteq \left(\left\{ z \in E | \left(\widehat{S}_1 \right)_z \cap I^\delta(u, v) \neq \emptyset \right\} \right) \right\} \quad (13)$$

In the final step, the output segmented image $O(u, v)$ is generated by concatenating $I^\theta(u, v)$ thrice to form the equivalent true color (RGB) image.

$$O(u, v) = I^\theta(u, v) \cap I^\theta(u, v) \cap I^\theta(u, v) \quad (14)$$

Algorithm II: Image Segmentation

Input: Input image $\rightarrow I(u, v)$

Output: Segmented image $\rightarrow O(u, v)$

Initialization: Thresholding value $\rightarrow \alpha$

Structuring Elements $\rightarrow S_1, S_2$

1: **procedure** :

2: Structuring Elements $\rightarrow S_1, S_2$

3: **if** $I(u, v) \geq \alpha$ **then**

4: $I^\beta(u, v) \leftarrow 1$

5: **else**

6: $I^\beta(u, v) \leftarrow 0$

7: **end if**

9: Dilation: $I^\gamma(u, v) \leftarrow I^\beta(u, v) \oplus S_1$

8: Opening operation: $I^\delta(u, v) \leftarrow I^\gamma(u, v) \circ S_2$

9: Fill hole: $I^{\delta_1}(u, v) \leftarrow I^\delta(u, v)$

10: Clear border: $I^{\delta_2}(u, v) \leftarrow I^{\delta_1}(u, v)$

11: Closing operation: $I^\theta(u, v) \leftarrow I^{\delta_2}(u, v) \bullet S_2$

12: Concatenation $O(u, v) \leftarrow I^\theta(u, v) \cap I^\theta(u, v) \cap I^\theta(u, v)$

13: **end procedure**

The MATLAB code used for the image processing task described above can be accessed through this Github link (<https://github.com/AjalaSunday/Neural-Networks-Fall-2021/blob/16ac1708307431272fc2c5c72ee25594d0a81446/PreprocessingPCFImages.m>).

Results and discussion

Model testing and evaluation metrics. The models were assessed for their performance by testing if pearl chain arrangements in a micrographs can be correlated to input voltages to find the model with the best performance using these four key performance metrics: Mean Absolute Error (MAE), Mean Relative Error (MRE), Mean Squared Error (MSE), R-squared, and Root Mean Square Error (RMSE)^{48,49,57,79}. They are mathematically expressed as given by Eqs. (15–20) where y , \hat{y} , and \bar{y} define the actual value, predicted value, and mean of the y values and n is the number of samples:

$$\text{MAE} = \frac{1}{n} \sum_i^n |y_i - \hat{y}_i| \quad (15)$$

$$\text{MRE} = \frac{1}{n} \sum_i^n \frac{|y_i - \hat{y}_i|}{y_i} \quad (16)$$

$$\text{MSE} = \frac{1}{n} \sum_i^n (y_i - \hat{y}_i)^2 \quad (17)$$

$$\text{R-squared} = 1 - \frac{\sum_i^n (y_i - \hat{y}_i)^2}{\sum_i^n (y_i - \bar{y})^2} \quad (18)$$

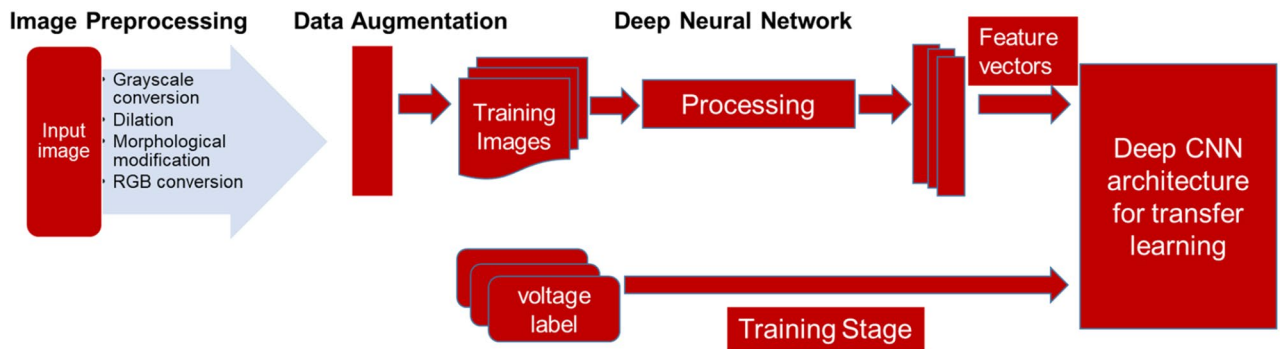


Figure 6. System overview showing the training phase of the models, the input image datasets were partitioned into 80% for the training and 20% for validation or testing.

$$\text{RMSE} = \sqrt{\frac{1}{n} \sum_i^n (y_i - \hat{y}_i)^2} \quad (19)$$

The prediction accuracy of the models can be defined is given by Eq. (20):

$$\text{Accuracy} = \frac{\sum_i^n |y_i - \hat{y}_i| < 0.5}{n} \quad (20)$$

The MAE assesses the average magnitude of errors in a group of predictions without taking into account their direction. It assesses the precision of continuous variables. MSE is often referred to as quadratic loss since the penalty is related to the square of the error rather than the error itself. When the error is squared, the outliers are given more weight, resulting in a smooth gradient for small errors. With an increase in error, MSE grows exponentially. The MSE value of a good model should be close to zero. RMSE is computed by taking the square root of MSE. RMSE is the more easily interpreted as it has the same units as the quantity. MAE, MSE, and RMSE can range from 0 to ∞ . The goodness of fit of a regression model is represented by a statistical measure called R-squared. The optimal R-squared value is 1. The closer the R-square value is to 1, the better the model fits.

Deep regression model for dielectrophoretic force estimation. After image processing and segmentation steps, the images are resized to fit the input layer dimension for each model (Table 2). These segmented image datasets are then augmented using augmentation procedures, such as randomly flipping them along the vertical axis and randomly translating them horizontally and vertically up to 30 pixels for training and validating the deep regression models (Fig. 6). Data augmentation keeps the networks from overfitting and ensure they adequately generalized. AlexNet, ResNet-50, MobileNetV2, and GoogLeNet were the four CNN architectures examined in this study. During the training phase of the models, the image datasets (2000 image samples each for yeast cells and PS microbeads) were partitioned into 80% (i.e. 1600 image samples each for yeast cells and PS microbeads) for the training and 20% (i.e. 400 image samples each for yeast cells and PS microbeads) for validation. The training was done in MATLAB R2021a, and the deep learning experiments were done with its Deep Learning toolbox. A DELL laptop with a five-core Intel 8th Generation processor served as our development system. The MATLAB code used for the deep learning described above is made available in this Githublink (<https://github.com/AjalaSunday/Neural-Networks-Fall2021/blob/16ac1708307431272fc2c5c72ee25594d0a81446/RegressionCode.m>).

By obtaining the R-squared, MAE, MRE, MSE, and RMSE values for the testing dataset base on the accuracy criteria, evaluation of the four architectures is performed. Tables 5 and 6 show the results achieved by the architectures and various optimizers in our experiments with yeast cells and microbeads respectively. As it can be seen in Table 5, all the models and optimization algorithms performed well above 95% based on the accuracy metric (also see Table S2). However, training the models on yeast cells dataset, ResNet-50 with RMSProp optimizer has the best validation RMSE of 0.0918 on test dataset, followed by the same ResNet-50 but with ADAM optimizer having a validation RMSE of 0.1241. This is also illustrated with the chart in Fig. 7. Figures S2 and S3 show the evolution of the validation RMSE for ResNet-50 with RMSProp optimizer and ResNet-50 with ADAM optimizer on the yeast cell dataset respectively while the regression lines are illustrated in Fig. 8a and b respectively.

On the microbeads dataset, as it can be seen in Table 6, all the models and optimization algorithms performed well above 90% based on the accuracy metric (also see Table S3 and S4). AlexNet with ADAM optimizer have the best validation RMSE of 0.1745 across all models on the PS microbeads dataset followed by ResNet-50 also with ADAM optimizer with validation RMSE 0.1869. This is also illustrated with the chart in Fig. 9. Figures S4 and S5 show the evolution of the validation RMSE for AlexNet with ADAM optimizer and ResNet-50 with ADAM optimizer on the PS microbeads dataset respectively while the regression lines are illustrated in Fig. 10a and b respectively. A look at the performances of adaptive learning rate optimization algorithms explored in this work, we found that ADAM has the least sum followed by RMSProp and then SGDM come last, having the highest sum of RMSE on both datasets as shown in Fig. 11a and b.

Architecture	Optimizer	MAE	MSE	RMSE	R-squared	Rank
AlexNet	ADAM	0.2089	0.0390	0.1976	0.995	8th
	SGDM	0.2009	0.0616	0.2483	0.993	11th
	RMSProp	0.1386	0.0403	0.2008	0.996	9th
ResNet-50	ADAM	0.0864	0.0154	0.1241	0.999	2nd
	SGDM	0.1609	0.0477	0.2185	0.994	10th
	RMSProp	0.0661	0.0084	0.0918	0.999	1st
MobileNetV2	ADAM	0.1072	0.0195	0.1398	0.999	4th
	SGDM	0.2099	0.0699	0.2645	0.993	12th
	RMSProp	0.1049	0.0216	0.1470	0.999	5th
GoogLeNet	ADAM	0.0781	0.0151	0.1230	0.999	3rd
	SGDM	0.1319	0.0290	0.1701	0.997	6th
	RMSProp	0.1038	0.0312	0.1768	0.997	7th

Table 5. CNN deep regression model performance on yeast cells for various architectures Significant values are in bold.

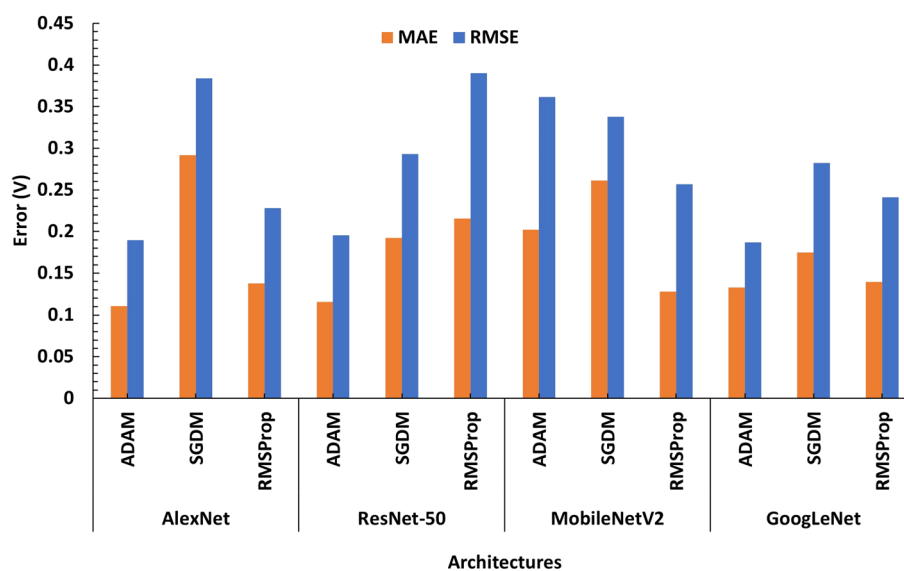


Figure 7. Model performance on yeast cells dataset for various architectures ResNet-50 with RMSPROP optimizer has the best validation RMSE of 0.0918 on test dataset, followed by the same ResNet-50 but with ADAM optimizer having a validation RMSE of 0.1241.

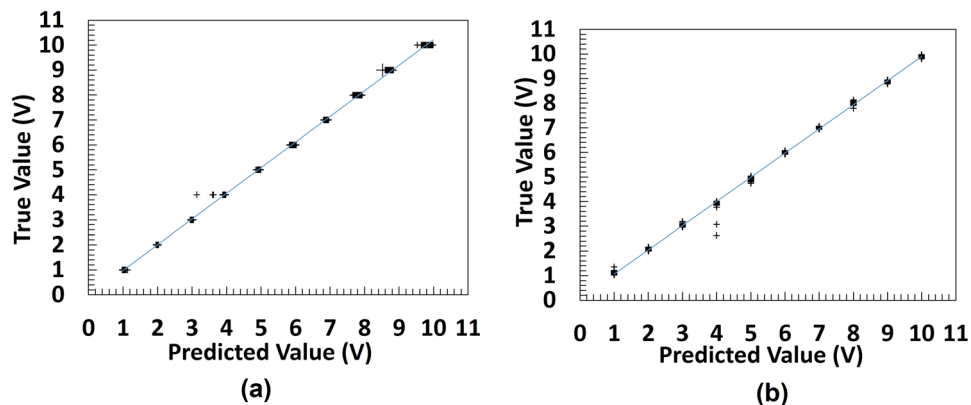


Figure 8. Line of best fit of the best deep regression models (a) ResNet-50 with RMSProp (b) ResNet-50 with ADAM.

Architecture	Optimizer	MAE	MSE	RMSE	R-squared	Rank
AlexNet	ADAM	0.1109	0.0305	0.1745	0.996	1st
	SGDM	0.2919	0.1474	0.3839	0.985	11th
	RMSProp	0.1376	0.0520	0.2281	0.994	4th
ResNet-50	ADAM	0.1329	0.0349	0.1869	0.998	2nd
	SGDM	0.1924	0.0859	0.2931	0.99	8th
	RMSProp	0.2156	0.1524	0.3904	0.987	12th
MobileNetV2	ADAM	0.2022	0.1307	0.3616	0.99	10th
	SGDM	0.2614	0.1142	0.3379	0.99	9th
	RMSProp	0.1277	0.0660	0.2570	0.994	6th
GoogLeNet	ADAM	0.1159	0.0383	0.1956	0.995	3rd
	SGDM	0.1750	0.0798	0.2825	0.992	7th
	RMSProp	0.1395	0.0580	0.2409	0.993	5th

Table 6. CNN Deep Regression Model Performance on microbeads for various Architectures. Significant values are in bold.

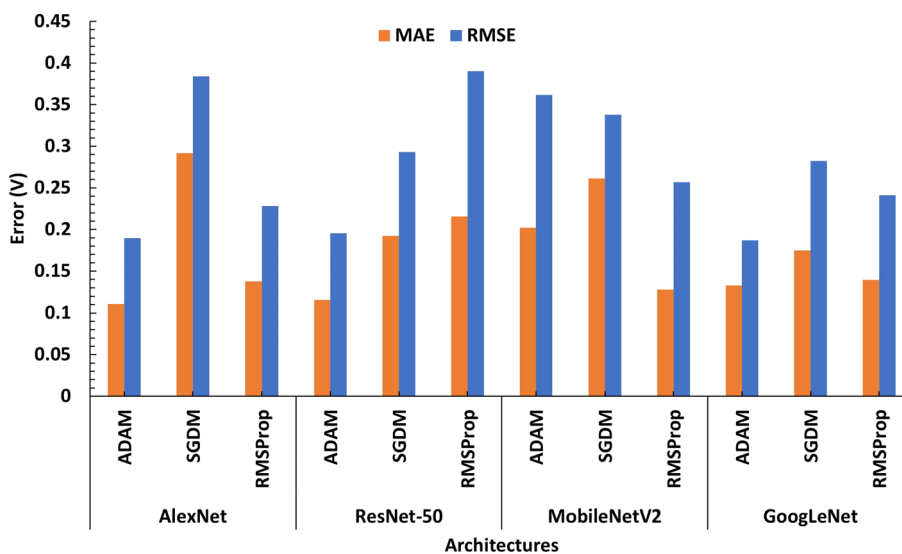


Figure 9. Model performance on microbeads dataset for various architectures AlexNet with ADAM optimizer has the best validation RMSE of 0.1745 on test dataset, followed by the same ResNet-50 but with ADAM optimizer having a validation RMSE of 0.1869.

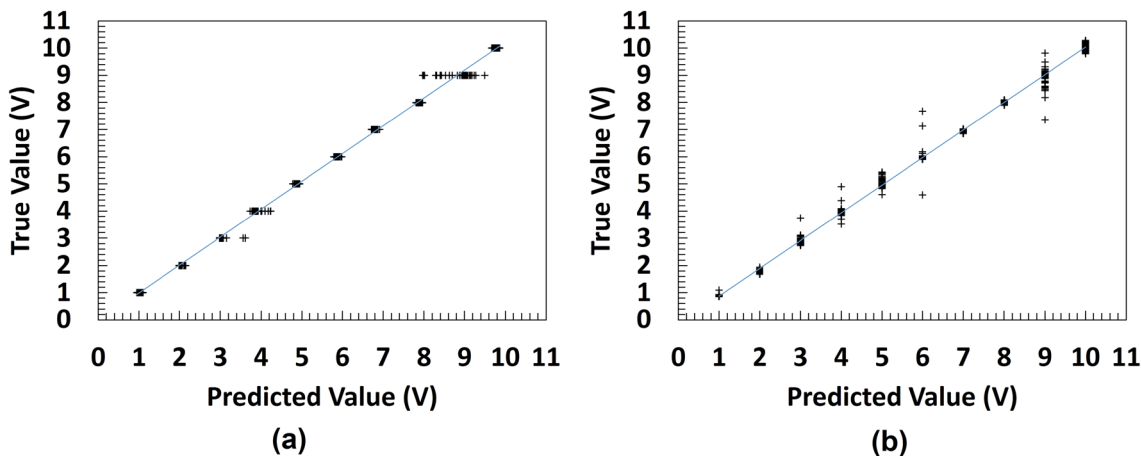


Figure 10. Line of best fit of the best deep regression models (a) AlexNet with ADAM (b) ResNet-50 with ADAM.

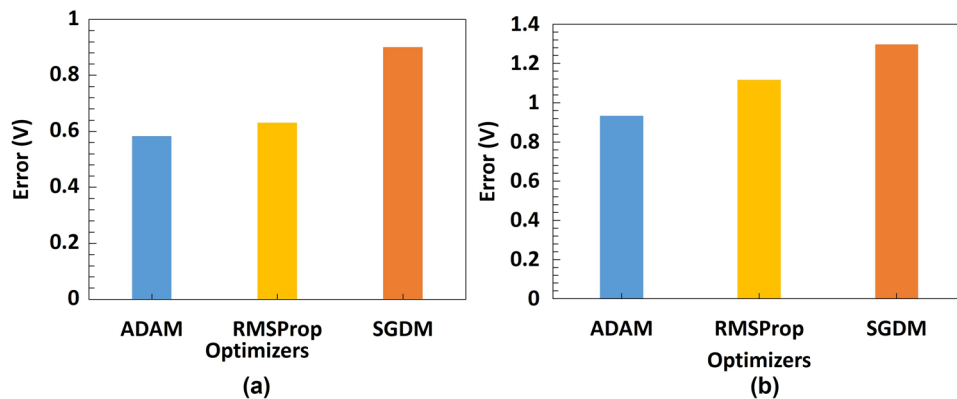


Figure 11. Sum of RMSE by Optimizer on (a) Yeast Cell Dataset (b) PS Microbead Dataset.

Conclusion

This paper presents an intelligent sensing framework capable of direct estimation of DEP force from pearl chain alignment of microparticles. We have tested the proposed models in an electrode-based dielectrophoretic system. The proposed deep regression models were extensively examined, and results were compared with conventional machine learning approaches. The intrinsic features of microparticle alignment like pearl chain length and count were extracted using image segmentation algorithms and used to generate training datasets. The results from the experiments show that the performance of the DL models proved to be optimal in terms of prediction accuracy and generalization ability compared to the ML models. ResNet-50 with RMSPROP gave the best performance, with a validation RMSE of 0.0918 on yeast cells while AlexNet with ADAM optimizer gave the best performance, with a validation RMSE of 0.1745 on microbeads. The regression model we developed can be extended to bio-sensing systems in order to estimate the variations in dielectric properties of microparticles.

Data availability

The dataset used for the current study is available on Dryad via this link. This is available to the reviewers but will be made available to the public after this article has been peer reviewed or upon request from the corresponding author.

Received: 23 March 2022; Accepted: 5 July 2022

Published online: 13 July 2022

References

- Pande, S., Khamparia, A. & Gupta, D. Feature selection and comparison of classification algorithms for wireless sensor networks. *J. Ambient. Intell. Humaniz. Comput.* <https://doi.org/10.1007/s12652-021-03411-6> (2021).
- Madhavan, M. V., Pande, S., Umekar, P., Mahore, T., & Kalyankar, D. Comparative Analysis of detection of email spam with the aid of machine learning approaches. In *IOP Conference Series: Materials Science and Engineering*, vol. 1022(1) 12113 <https://doi.org/10.1088/1757-899x/1022/1/012113> (2021).
- Dharmale, S. G., Gomase, S. A., Pande, S. Comparative analysis on machine learning methodologies for the effective usage of medical WSNs. In *Proceedings of Data Analytics and Management* 441–457 (2022).
- Sirohi, M., Lall, M., Yenishetti, S., Panat, L. & Kumar, A. Development of a Machine learning image segmentation-based algorithm for the determination of the adequacy of Gram-stained sputum smear images. *Med. J. Armed Forces India* <https://doi.org/10.1016/J.MJAFI.2021.09.012> (2021).
- Prinyakupt, J. & Pluempitiwiriyaew, C. Segmentation of white blood cells and comparison of cell morphology by linear and naive Bayes classifiers. *Biomed. Eng. Online* **14**(1), 1–19. <https://doi.org/10.1186/S12938-015-0037-1/TABLES/8> (2015).
- Petrović, N., Moyà-Alcover, G., Jaume-i-Capó, A. & González-Hidalgo, M. Sickle-cell disease diagnosis support selecting the most appropriate machine learning method: Towards a general and interpretable approach for cell morphology analysis from microscopy images. *Comput. Biol. Med.* **126**, 104027. <https://doi.org/10.1016/J.COMPBIOMED.2020.104027> (2020).
- Pratapa, A., Doron, M. & Caicedo, J. C. Image-based cell phenotyping with deep learning. *Curr. Opin. Chem. Biol.* **65**, 9–17. <https://doi.org/10.1016/J.CBPA.2021.04.001> (2021).
- Hussain, S. *et al.* High-content image generation for drug discovery using generative adversarial networks. *Neural Netw.* **132**, 353–363. <https://doi.org/10.1016/J.NEUNET.2020.09.007> (2020).
- Liu, L., Xie, C., Chen, B. & Wu, J. Numerical study of particle chains of a large number of randomly distributed DEP particles using iterative dipole moment method. *J. Chem. Technol. Biotechnol.* **91**(4), 1149–1156. <https://doi.org/10.1002/JCTB.4700> (2016).
- Pohl, H. A. & Crane, J. S. Dielectrophoresis of cells. *Biophys. J.* **11**(9), 711–727. [https://doi.org/10.1016/S0006-3495\(71\)86249-5](https://doi.org/10.1016/S0006-3495(71)86249-5) (1971).
- Washizu, M. & Jones, T. B. Generalized multipolar dielectrophoretic force and electrorotational torque calculation. *J. Electrostat.* **38**(3), 199–211. [https://doi.org/10.1016/S0304-3886\(96\)00025-3](https://doi.org/10.1016/S0304-3886(96)00025-3) (1996).
- Ajala, S., Jalajamony, H. M. & Fernandez, R. E. Deep learning based image analysis of Pearl chain formation in a dielectrophoretic system. In *ECSSMA*, vol. 2021(35) 1962–1962 <https://doi.org/10.1149/MA2021-02351962MTGABS> (2021).
- Zhao, Y., Brcka, J., Faguet, J. & Zhang, G. Elucidating the mechanisms of two unique phenomena governed by particle-particle interaction under DEP: Tumbling motion of pearl chains and alignment of ellipsoidal particles. *Micromachines* **2018** **9**(6), 279. <https://doi.org/10.3390/MI9060279> (2018).
- Daniel, J. *et al.* Pearl-chain formation of discontinuous carbon fiber under an electrical field. *J. Manuf. Mater. Process.* **2017** **1**(2), 22. <https://doi.org/10.3390/JMMP1020022> (2017).

15. Zhao, Y., Hodge, J., Brcka, J., Faguet, J., Lee, E. & Zhang, G. Effect of electric field distortion on particle-particle interaction under DEP excerpt from the proceedings of the 2013 COMSOL Conference in Boston.
16. Kanagasabapathi, T., Backhouse, T. & Kaler, K. Dielectrophoresis (DEP) of Cells and Microparticle in PDMS Microfluidic Channels. In *NSTI-Nanotech*, vol. 1 (2004).
17. Ajala, S., Jalajamony, H. M. & Fernandez, R. E. Deep-learning based estimation of dielectrophoretic force. *Micromachines* **2022** *13*(1), 41. <https://doi.org/10.3390/M113010041> (2021).
18. Li, M., Fei, F., Qu, Y., Dong, Z., Li, W. J. & Wang, Y. Theoretical analysis based on particle electro-mechanics for Au Pearl Chain Formation. In *2007 7th IEEE International Conference on Nanotechnology—IEEE-NANO 2007, Proceedings*, 1217–1221 <https://doi.org/10.1109/NANO.2007.4601402>(2007).
19. Wu, C. *et al.* A planar dielectrophoresis-based chip for high-throughput cell pairing. *Lab Chip* **17**(23), 4008–4014. <https://doi.org/10.1039/C7LC01082F> (2017).
20. Fernandez, R. E., Koklu, A., Mansoorifar, A. & Beskok, A. Platinum black electrodeposited thread based electrodes for dielectrophoretic assembly of microparticles. *Biomicrofluidics* **10**(3), 033101. <https://doi.org/10.1063/1.4946015> (2016).
21. Fernandez, R. E., Rohani, A., Farmehini, V. & Swami, N. S. Review: microbial analysis in dielectrophoretic microfluidic systems. *Anal. Chim. Acta* **966**, 11. <https://doi.org/10.1016/J.ACA.2017.02.024> (2017).
22. Benhal, P., Quashie, D., Kim, Y. & Ali, J. Insulator based dielectrophoresis: micro, nano, and molecular scale biological applications. *Sensors* **2020** *20*(18), 5095. <https://doi.org/10.3390/S20185095> (2020).
23. Malekanfard, A., Beladi-Behbahani, S., Tzeng, T. R., Zhao, H. & Xuan, X. AC insulator-based dielectrophoretic focusing of particles and cells in an ‘infinite’ microchannel. *Anal. Chem.* **93**(14), 5947–5953. <https://doi.org/10.1021/ACS.ANALCHEM.1C00697> (2021).
24. Shafiee, H., Sano, M. B., Henslee, E. A., Caldwell, J. L. & Davalos, R. V. Selective isolation of live/dead cells using contactless dielectrophoresis (cDEP). *Lab Chip* **10**(4), 438–445. <https://doi.org/10.1039/B920590J> (2010).
25. Ho, C. T., Lin, R. Z., Chang, W. Y., Chang, H. Y. & Liu, C. H. Rapid heterogeneous liver-cell on-chip patterning via the enhanced field-induced dielectrophoresis trap. *Lab Chip* **6**(6), 724–734. <https://doi.org/10.1039/B602036D> (2006).
26. Jen, C. P. & Chen, T. W. Selective trapping of live and dead mammalian cells using insulator-based dielectrophoresis within open-top microstructures. *Biomed. Microdevices* **11**(3), 597–607. <https://doi.org/10.1007/S10544-008-9269-1> (2009).
27. Shafiee, H., Caldwell, J. L., Sano, M. B. & Davalos, R. V. Contactless dielectrophoresis: a new technique for cell manipulation. *Biomed. Microdevices* **11**(5), 997–1006. <https://doi.org/10.1007/S10544-009-9317-5> (2009).
28. Guérin, N. *et al.* Helical dielectrophoretic particle separator fabricated by conformal spindle printing. *J. Biomed. Sci. Eng.* **7**(9), 641–650. <https://doi.org/10.4236/JBISE.2014.79064> (2014).
29. Yang, L., Banada, P. P., Bhunia, A. K. & Bashir, R. Effects of Dielectrophoresis on growth, viability and immuno-reactivity of *Listeria monocytogenes*. *J. Biol. Eng.* **2**(1), 1–14. <https://doi.org/10.1186/1754-1611-2-6/TABLES/1> (2008).
30. Miled, M. A., Massicotte, G. & Sawan, M. Dielectrophoresis-based integrated lab-on-chip for nano and micro-particles manipulation and capacitive detection. *IEEE Trans. Biomed. Circuits Syst.* **7**(4), 557. <https://doi.org/10.1109/TBCAS.2013.2271727> (2012).
31. Rahman, N. A., Ibrahim, F. & Yafouz, B. Dielectrophoresis for biomedical sciences applications: a review. *Sensors* <https://doi.org/10.3390/s17030449> (2017).
32. Demircan, Y., Özgür, E. & Külah, H. Dielectrophoresis: applications and future outlook in point of care. *Electrophoresis* **34**(7), 1008–1027. <https://doi.org/10.1002/ELPS.201200446> (2013).
33. Sapsford, K. E., Taitt, C. R., Loo, N. & Ligler, F. S. Biosensor detection of botulinum toxoid A and staphylococcal enterotoxin B in food. *Appl. Environ. Microbiol.* **71**(9), 5590. <https://doi.org/10.1128/AEM.71.9.5590-5592.2005> (2005).
34. Khoshmanesh, K. *et al.* Interfacing cell-based assays in environmental scanning electron microscopy using dielectrophoresis. *Anal. Chem.* **83**(8), 3217–3221. <https://doi.org/10.1021/AC2002142> (2011).
35. Pohl, H. A. *Dielectrophoresis: the behavior of neutral matter in nonuniform electric fields* (Cambridge University Press, 1978).
36. Oh, M., Jayasooriya, V., Woo, S. O., Nawarathna, D. & Choi, Y. Selective manipulation of biomolecules with insulator-based dielectrophoretic tweezers. *ACS Appl. Nano Mater.* **3**(1), 797–805 (2020).
37. Ettehad, H. M., Yadav, R. K., Guha, S. & Wenger, C. Towards CMOS integrated microfluidics using dielectrophoretic immobilization. *Biosensors (Basel)* <https://doi.org/10.3390/BIOS9020077> (2019).
38. Manaresi, N. *et al.* A CMOS chip for individual cell manipulation and detection. *IEEE J. Solid-State Circuits* **38**(12), 2297–2305. <https://doi.org/10.1109/JSSC.2003.819171> (2003).
39. Kurgan, E. Comparison of different force calculation methods in DC dielectrophoresis. *Electrotech. Rev.* **88**(8), 11–14 (2012).
40. C. Xie, B. Chen, L. Liu, H. Chen, and J. Wu, Iterative dipole moment method for the interaction of multiple dielectrophoretic particles in an AC electrical field. *Eur. J. Mech. B/Fluids* **58**, 50–58. <https://doi.org/10.1016/j.euromechflu.2016.03.003> (2016).
41. Ai, Y., Beskok, A., Gauthier, D. T., Joo, S. W. & Qian, S. DC electrokinetic transport of cylindrical cells in straight microchannels. *Biomicrofluidics* **3**(4), 44110. <https://doi.org/10.1063/1.3267095> (2009).
42. Ai, Y., Joo, S. W., Jiang, Y., Xuan, X. & Qian, S. Transient electrophoretic motion of a charged particle through a converging-diverging microchannel: effect of direct current-dielectrophoretic force. *Electrophoresis* **30**(14), 2499–2506. <https://doi.org/10.1002/ELPS.200800792> (2009).
43. Ai, Y. & Qian, S. DC dielectrophoretic particle-particle interactions and their relative motions. *J. Colloid Interface Sci.* **346**(2), 448–454. <https://doi.org/10.1016/J.JCIS.2010.03.003> (2010).
44. Ai, Y., Zeng, Z. & Qian, S. Direct numerical simulation of AC dielectrophoretic particle-particle interactive motions. *J. Colloid Interface Sci.* **417**, 72–79. <https://doi.org/10.1016/J.JCIS.2013.11.034> (2014).
45. Liu, L. *et al.* Iterative dipole moment method for calculating dielectrophoretic forces of particle-particle electric field interactions. *Appl. Math. Mech.* **36**(11), 1499–1512. <https://doi.org/10.1007/S10483-015-1998-7> (2015).
46. Su, Y. H. *et al.* Quantitative dielectrophoretic tracking for characterization and separation of persistent subpopulations of *Cryptosporidium parvum*. *Analyst* **139**(1), 66–73. <https://doi.org/10.1039/c3an01810e> (2013).
47. Lathuilière, S., Mesejo, P., Alameda-Pineda, X. & Horaud, R. A comprehensive analysis of deep regression. *IEEE Trans. Pattern Anal. Mach. Intell.* **42**(9), 2065–2081. <https://doi.org/10.1109/TPAMI.2019.2910523> (2018).
48. Xue, Y., Ray, Y., Hugh, J. & Bigras, J. Cell Counting by regression using convolutional neural network. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 9913 LNCS 274–290 https://doi.org/10.1007/978-3-319-46604-0_20 (2016).
49. Xie, Y., Xing, F., Kong, X., Su, H. & Yang, L. Beyond classification: structured regression for robust cell detection using convolutional neural network. *Med. Image Comput. Comput. Assist. Interv.* **9351**, 358. https://doi.org/10.1007/978-3-319-24574-4_43 (2015).
50. Fanelli, G., Gall, J. & Van Gool, L. Real time head pose estimation with random regression forests. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 617–624 <https://doi.org/10.1109/CVPR.2011.5995458> (2011).
51. Zhu, X. & Ramanan, D. Face detection, pose estimation, and landmark localization in the wild. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2879–2886 <https://doi.org/10.1109/CVPR.2012.6248014>(2012).
52. Sabottke, C. F., Breaux, M. A. & Spieler, B. M. Estimation of age in unidentified patients via chest radiography using convolutional neural network regression. *Emergency Radiol.* **2020** *27*(5), 463–468. <https://doi.org/10.1007/S10140-020-01782-5> (2020).

53. Niu, Z., Zhou, M., Wang, L., Gao, X. & Hua, G. Ordinal regression with multiple output CNN for age estimation. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2016–December 4920–4928 <https://doi.org/10.1109/CVPR.2016.532> (2016).
54. Xie, C., Chen, B., Ng, C. O., Zhou, X. & Wu, J. Numerical study of interactive motion of dielectrophoretic particles. *undefined* **49**, 208–216. <https://doi.org/10.1016/J.EUROMECHFLU.2014.08.007> (2015).
55. Techaumnat, B., Eua-Arporn, B. & Takuma, T. Calculation of electric field and dielectrophoretic force on spherical particles in chain. *J. Appl. Phys.* **95**(3), 1586–1593. <https://doi.org/10.1063/1.1637138> (2004).
56. Ogbi, A., Nicolas, L., Perrussel, R., & Voyer, D. Calculation of DEP force on spherical particle in non-uniform electric fields. In *Numélec 2012* 180 (2012).
57. Tudose, A. M. *et al.* Short-term load forecasting using convolutional neural networks in COVID-19 context: the Romanian case study. *Energies* **2021** **14**(13), 4046. <https://doi.org/10.3390/EN14134046> (2021).
58. LeCun, Y., Kavukcuoglu, K., & Farabet, C. Convolutional networks and applications in vision. In *ISCAS 2010—2010 IEEE International Symposium on Circuits and Systems: Nano-Bio Circuit Fabrics and Systems*, 253–256 <https://doi.org/10.1109/ISCAS.2010.5537907> (2010).
59. Abdolhoseini, M., Kluge, M. G., Walker, F. R. & Johnson, S. J. Segmentation of heavily clustered nuclei from histopathological images. *Sci. Rep.* **2019** **9**(1), 1–13. <https://doi.org/10.1038/s41598-019-38813-2> (2019).
60. Zhou, S. *et al.* High-resolution encoder-decoder networks for low-contrast medical image segmentation. *IEEE Trans. Image Process.* **29**, 461–475. <https://doi.org/10.1109/TIP.2019.2919937> (2020).
61. Ringenberg, J. *et al.* Automated segmentation and reconstruction of patient-specific cardiac anatomy and pathology from in vivo MRI*. *MeScT* **23**(12), 125405. <https://doi.org/10.1088/0957-0233/23/12/125405> (2012).
62. Pyo, J. C. *et al.* A convolutional neural network regression for quantifying cyanobacteria using hyperspectral imagery. *Remote Sens. Environ.* **233**, 111350. <https://doi.org/10.1016/J.RSE.2019.111350> (2019).
63. Tsochatzidis, L., Costaridou, L. & Pratikakis, I. Deep learning for breast cancer diagnosis from mammograms—a comparative study. *J. Imaging* **2019** **5**(3), 37. <https://doi.org/10.3390/JIMAGING5030037> (2019).
64. Yamlome, P., Akwaboah, A. D., Marz, A. & Deo, M. Convolutional neural network based breast cancer histopathology image classification. In *Proceedings of the Annual International Conference of the IEEE Engineering in Medicine and Biology Society, EMBS*, vol. 2020–July 1144–1147 <https://doi.org/10.1109/EMBC44109.2020.9176594> (2020).
65. Ossama, A.-H. *et al.* Convolutional neural networks for speech recognition. *IEEE/ACM Trans. Audio Speech. Lang. Process.* <https://doi.org/10.1109/TASLP.2014.2339736> (2014).
66. Elhoseiny, M., Huang, S. & Elgammal, A. Weather classification with deep convolutional neural networks. In *Proceedings—International Conference on Image Processing, ICIP*, vol. 2015–December, 3349–3353 <https://doi.org/10.1109/ICIP.2015.7351424> (2015).
67. Krizhevsky, A., Sutskever, I. & Hinton, G. E. ImageNet classification with deep convolutional neural networks. *Commun. ACM* <https://doi.org/10.1145/3065386> (2017).
68. Sandler, M., Howard, A., Zhu, M., Zhmoginov, A. & Chen, L. C. MobileNetV2: inverted residuals and linear bottlenecks. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 4510–4520 <https://doi.org/10.1109/CVPR.2018.00474> (2018).
69. Adetiba, E. *et al.* LeafsnpNet: an experimentally evolved deep learning model for recognition of plant species based on leafsnap image dataset. *J. Comput. Sci.* **17**(3), 349–363. <https://doi.org/10.3844/JCSSP.2021.349.363> (2021).
70. Simonyan, K. & Zisserman, A. Very deep convolutional networks for large-scale image recognition. In: *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings* (2014).
71. He, K., Zhang, X., Ren, S., & Sun, J. Deep residual learning for image recognition. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Vol. 2016–December, 770–778 <https://doi.org/10.1109/CVPR.2016.90> (2015).
72. Ruder, S. An overview of gradient descent optimization algorithms. (2016).
73. “Gradient Descent With RMSProp from Scratch.” <https://machinelearningmastery.com/gradient-descent-with-rmsprop-from-scratch/> (accessed Jan. 07, 2022).
74. “Stochastic Gradient Descent with momentum | by Vitaly Bushaev | Towards Data Science.” <https://towardsdatascience.com/stochastic-gradient-descent-with-momentum-a84097641a5d> (accessed Jan. 07, 2022).
75. Petushi, S., Garcia, F. U., Haber, M. M., Katsinis, C. & Tozeren, A. Large-scale computations on histology images reveal grade-differentiating parameters for breast cancer. *BMC Med. Imaging* **6**(1), 1–11. <https://doi.org/10.1186/1471-2342-6-14/FIGURES/10> (2006).
76. Nazir, I. *et al.* Efficient Pre-processing and segmentation for lung cancer detection using fused CT images. *Electronics* **2022** **11**(1), 34. <https://doi.org/10.3390/ELECTRONICS11010034> (2021).
77. Senthilkumar, N. & Thimmiraja, J. An illustrative analysis of mathematical morphology operations for MRI brain images. *Int. J. Comput. Sci. Inform. Technol.* **5**(3), 2684–2688 (2014).
78. Fang, Z., Junpeng, Z., Zhumadian, H. & Yulei, M. Medical image processing based on mathematical morphology. In *The 2nd International Conference on Computer Application and System Modeling* (2012).
79. Kalampokas, T., Vrochidou, E., Papakostas, G. A., Pachidis, T. & Kaburlasos, V. G. Grape stem detection using regression convolutional neural networks. *Comput. Electron. Agric.* **186**, 106220. <https://doi.org/10.1016/J.COMPAG.2021.106220> (2021).

Acknowledgements

Support from National science foundation grant #2100930 is acknowledged.

Author contributions

R.F. designed the experiments, sensors, developed the machine learning modules and co-wrote the manuscript. S.A. developed the deep learning analysis and co-wrote the manuscript. H.J. designed and fabricated the sensor and conducted experiments. M.N. and P.M. conducted the machine-learning analysis including image processing. All authors participated in completing the manuscript.

Competing interests

The authors declare no competing interests.

Additional information

Supplementary Information The online version contains supplementary material available at <https://doi.org/10.1038/s41598-022-16114-5>.

Correspondence and requests for materials should be addressed to R.E.F.

Reprints and permissions information is available at www.nature.com/reprints.

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

© The Author(s) 2022