International Journal of Bioprinting

REVIEW ARTICLE

Machine learning boosts three-dimensional bioprinting

Hongwei Ning[1], Teng Zhou[2]*, Sang Woo Joo[3]*

[1]College of Information and Network Engineering, Anhui Science and Technology University, Bengbu, Anhui, China

[2]Mechanical and Electrical Engineering College, Hainan University, Haikou, Hainan, China

[3]School of Mechanical Engineering, Yeungnam University, Gyeongsan, Korea

(This article belongs to the *Special Issue: Advances in 3D bioprinting for regenerative medicine and drug screening*)

Abstract

Three-dimensional (3D) bioprinting is a computer-controlled technology that combines biological factors and bioinks to print an accurate 3D structure in a layer-by-layer fashion. 3D bioprinting is a new tissue engineering technology based on rapid prototyping and additive manufacturing technology, combined with various disciplines. In addition to the problems in *in vitro* culture process, the bioprinting procedure is also afflicted with a few issues: (1) difficulty in looking for the appropriate bioink to match the printing parameters to reduce cell damage and mortality; and (2) difficulty in improving the printing accuracy in the printing process. Data-driven machine learning algorithms with powerful predictive capabilities have natural advantages in behavior prediction and new model exploration. Combining machine learning algorithms with 3D bioprinting helps to find more efficient bioinks, determine printing parameters, and detect defects in the printing process. This paper introduces several machine learning algorithms in detail, summarizes the role of machine learning in additive manufacturing applications, and reviews the research progress of the combination of 3D bioprinting and machine learning in recent years, especially the improvement of bioink generation, the optimization of printing parameter, and the detection of printing defect.

Keywords: Bioprinting; Additive manufacturing; K-nearest neighbor; Long short-term memory; Ensemble learning

***Corresponding authors:**
Sang Woo Joo
(swjoo@yu.ac.kr)

Teng Zhou
(zhouteng@hainanu.edu.cn)

Citation: Ning H, Zhou T, Joo SW, 2023, Machine learning boosts three-dimensional bioprinting. *Int J Bioprint,* 9(4): 739. https://doi.org/10.18063/ijb.739

Received: February 03, 2023
Accepted: March 06, 2023
Published Online: April 27, 2023

1. Introduction

Three-dimensional (3D) printing technology, which is also called additive manufacturing, is a branch of rapid prototyping technology. It is a manufacturing technology that accumulates materials layer by layer and solidifies them to obtain solid finished products[1,2]. The 3D model obtained by computer rendering or scanning is first discretized into a stack of parallel layers by slicing software. Then, through the numerical control system, spraying, extrusion, hot melting, laser, and other methods, the filament-like, liquid or powdered plastic, ceramic, metal, and other materials are positioned, scanned, and stacked layer by layer. Finally, the printed solid product is obtained[3]. In recent years, 3D printing technology has attracted much attention because of its

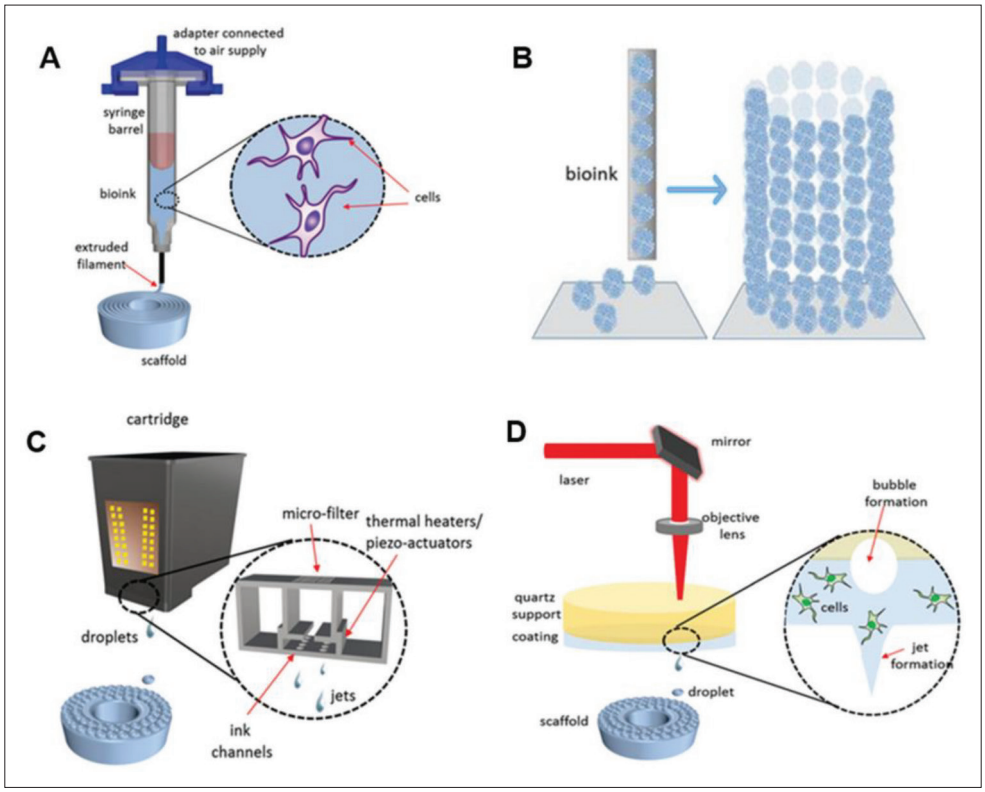

Figure 1. Type of 3D bioprinting. (A) Extrusion-based bioprinting. (B) Scaffold-free bioprinting. (C) Inkjet bioprinting. (D) Laser-assisted bioprinting. Reprinted from ref. [5] under the terms of the Creative Commons CC-BY license.

theoretical versatility, diverse materials, and abilities of accurately reproduce complex models. The application of 3D printing technology has also penetrated from the initial manufacturing industry to various industries, including the biomedical field, especially in fabricating human tissues and organs, showing great value and application prospects[4].

3D bioprinting provides precise control for the manufacture of scaffolds with natural extracellular matrix properties. Common types of printing include inkjet bioprinting, laser-assisted bioprinting, extrusion-based bioprinting, and scaffold-free bioprinting[5] (Figure 1). The bioinks of inkjet bioprinting are mainly composed of biological materials and cells. By changing the printer's parameters, droplets of different sizes can be intermittently sprayed from the nozzle. The method could guarantee high print speed and outstanding cell activity while ensuring lower cost. Laser-assisted bioprinting mainly offers the driving force for the bioink printing by focusing the laser on the printer substrate[6]. Compared with other printing technologies, laser-assisted bioprinting can guarantee ideal cell viability with a survival rate of up to 95%. Extrusion-based 3D bioprinting is currently the most widely utilized technology in the scope of biological manufacturing. It mainly uses the air stress or mechanical driving force of the piston to make the composite bioink of biomaterials, cells or growth factors continuously extruded from the needle nozzle in the form of filaments. The 3D lithography printing system prints a 3D structure by laser, ultraviolet or visible light, or applies selective light and laser light curing of biological materials to print 3D structures[7,8].

For 3D bioprinting, the final printing effect is determined by a variety of printing parameters. For example, parameters such as the speed set by the nozzle, the pressure, and the temperature used for extruding materials have a direct impact on cell activities. To secure the survival of the cells in the printed structure, it is usually necessary to mix the cells with the bioink before printing[9]. Bioink is used as a carrier in the field of 3D bioprinting. It is traditionally applied to wrap cells in it, to ensure the average growth and survival of the cells. Bioinks provide cells with a culture environment similar to an extracellular matrix, so they are often used as cell carriers in 3D bioprinting. Bioink is mainly divided into natural materials and synthetic materials. Biological manufacturing requires that bioink has good biological properties and printing performance. Therefore, it needs to be selected according to the specific manufacturing needs of bioink[10].

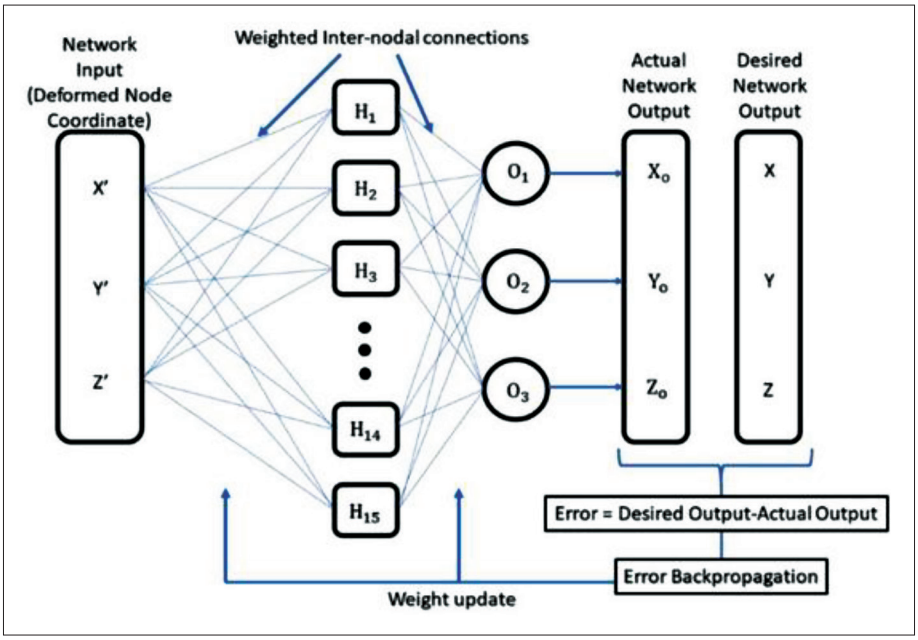

Figure 2. An error compensation method used in additive manufacturing. Reprinted from ref. [17] under the terms of the Creative Commons CC-BY license.

Although 3D bioprinting has the potential to be applied in the regeneration strategy, many problems concerning the printing process and the materials that can be used for bioprinting are waiting to be overcome[11,12]. The challenges of the printing procedure include printing accuracy, printing speed, and compatibility of the printing procedures with cells. The most significant limitation of bioprinting technology is that the printed tissue structures do not resemble the natural tissues and organs. Most of the time, the current bioprinting methods can only achieve structural design and fabrication but the fabricated structures can hardly function like how their natural counterparts do.

After analyzing relevant features in existing data, machine learning can be employed to process new data. Machine learning, with its rich experience, is more adept at dealing with new problems[13]. In recent years, there have been a lot of investigations targeting at combining machine learning with 3D bioprinting, and favorable outcomes have been obtained[14,15]. In this paper, the recent work about 3D bioprinting in bioink preparation, parameter optimization, and defect detection through machine learning are summarized. The paper is organized as follows: First, the applications of machine learning in additive manufacturing are listed, shedding the light on the application of machine learning technology in 3D bioprinting. Then, the basic principles of machine learning algorithms such as K-nearest neighbor, back propagation neural network, convolutional neural network, long short-time memory, and integrated learning are shown. Next, the successful applications of machine learning technology in bioink preparation, printing parameter optimization, and defect detection are summarized in the subsequent sections.

2. Additive manufacturing with machine learning

3D bioprinting is a new tissue engineering technology based on rapid prototyping and additive manufacturing technology, mostly involving multiple disciplines. Additive manufacturing is the general name of a method used to construct 3D object from computer-aided design model, while bioprinting is a branch of this method that is predominantly used in fabricating biological constructs[16,17] (Figure 2). By reviewing and summarizing the instances of the combination of additive manufacturing and machine learning, it is helpful to understand the advantages of the combination of machine learning and bioprinting more profoundly and further expand the application of machine learning in bioprinting[18,19].

At present, the product development of additive manufacturing is not mature enough, and the design rules also need more in-depth research. Ko *et al.*[20] designed a knowledge reasoning structure with a decision tree algorithm. The dataset for model training was derived from the additive manufacturing case of laser powder fusion. The trained model was verified on the test data of the National

Institute of Standards and Measurement Technology, and the results demonstrated that it had great significance. Li *et al.*[21] surveyed a scheme to uncover the relationship between the design method of additive manufacturing and the roughness of the final product. Random forest, decision tree, and support vector regression algorithms were integrated into the scheme. In the actual printing case, the roughness data of the printed product was collected based on multiple sensors, and a dataset was constituted for training the ensemble learning algorithm. The results verified that the scheme could reveal the relationship between bioprinting design and final print roughness. Zhu *et al.*[22] presented a mode to obtain the shape deviation between the theoretical design of printing and the actual prints. The mode included the transformation perspective algorithm and Gaussian process algorithm. The transformation perspective algorithm built the theory model, while the Gaussian process algorithm learned shape deviation data and predicted shape deviation. The data in the dataset were obtained by measuring the printed product with a 3D laser scanner. Explanatory cases demonstrated the effect of the mode. To obtain better results, the group considered adding more parameters to the framework. In order to assess the print performance corresponding to bioprinting design, Jiang *et al.*[23] conducted in-depth research based on deep neural network algorithms. Machine learning could find the relationship between bioprinting design space and performance space, and deep neural network algorithms had advantages in input–output relationship mapping. The simulated data between stress and strain responses formed the dataset for training a deep neural network. The case of ankle scaffold bioprinting indicated that the approach was practical.

Machine learning also plays a vital role in the defect detection of additive manufacturing. Ghayoomi Mohammadi *et al.*[24] applied various machine learning algorithms to achieve real-time defect detection during laser powder additive printing. The data processed by the machine learning algorithm and the training data in the dataset were both generated by the continuous monitoring of the printing by the acoustic emission sensor. The k-means clustering technique marked the acoustic data, and the neural network improved the precision of the data. The principal component analysis was performed to detect defect in real time, and the Gaussian mixture model was conducted to determine defect. The example of tool steel printing proved that the method was reliable. Caggiano *et al.*[25] combined machine learning with image processing to detect printing defects online. The machine learning algorithm used in the procedure was a deep convolutional neural network, and the processed image came from the layered image in the laser melting process. The experiment proved that the technology was effective and provided a strong guarantee for ensuring the quality of printed products. Gobert *et al.*[26] combined the linear support vector machine algorithm with the high-resolution layered image to detect defects during the printing period. In the printing process, multiple high-definition photos were taken by high-resolution digital single-lens reflex cameras for each layer to construct a dataset. During the printing process, each layer of the printed product was continuously imaged by computed tomography, and the image was processed with the trained model to detect problems in time. The cross-validation implementation confirmed that the accuracy of the method could reach at least 80%. Li *et al.*[27] applied a variety of machine learning algorithms to detect geometric faults in the printing process. The data in the dataset were not the data in the objective case, but the artificially synthesized 3D defect data, to save the training time and related costs. K-nearest neighbor, bagging of trees, gradient boosting, random forest, and support vector machine algorithms were all employed in the research. Experimental data showed that the two algorithms, bagging of trees and random forest, were the best.

In line with the above-mentioned instances, machine learning algorithms have achieved significant advances in the design and defect detection in the field of additive manufacturing. As a special branch of additive manufacturing, 3D bioprinting is likely to encounter similar problems in product design and defect detection. Thus, a summary regarding the advances of machine learning in additive manufacturing can provide a very good guidance on the design and defect detection of 3D-bioprinted products. A timely review in this regard can shed light on the solutions to the existing problems in 3D bioprinting, and promote the expansion of the utilization of machine learning in 3D bioprinting as well as the rapid development of 3D bioprinting. Of course, there are many applications of machine learning in additive manufacturing beyond product design and defect detection.

3. Theory of several machine learning algorithms

3.1. K-nearest neighbor

K-nearest neighbor method is a supervised learning algorithm in machine learning. Each data in the dataset should be labeled in advance. The k-nearest neighbor algorithm is one of the most concise algorithms in the machine learning field, which can be applied in both data classification and data regression[28]. The algorithm needs to store all the data, traversing all the data each time a prediction is made. At the same time, the algorithm does

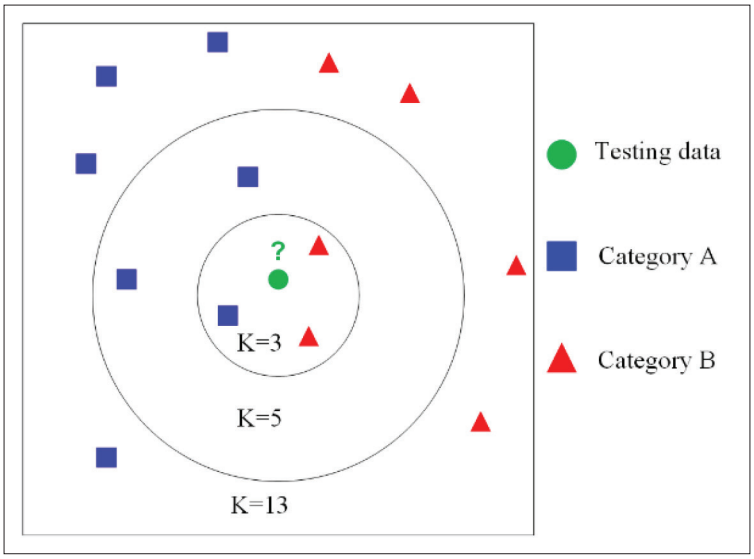

Figure 3. Schematic figure of the k-nearest neighbor algorithm. Reprinted from ref. [32] under the terms of the Creative Commons CC-BY license.

little or no processing of the data. The k-nearest neighbor algorithm has no explicit learning or training procedure, which belongs to lazy learning. When there is little or no prior information about data construction, the k-nearest neighbor algorithm is a good option[29].

The method owns a straightforward principle: when test instances are classified, the training dataset is scanned first to find *k* training samples that are most similar to the test dataset. Then, the type of the test instance is evaluated by voting according to the category of this sample, or weighted voting will be conducted on the correlation between each instance and the test pieces. If the output is required in the form of the probability of the test instance corresponding to each type, it can be estimated by the distribution of the number of instances with different categories in each dataset[30].

3.1.1. Algorithm implementation process

After the training and test dataset are ready, the k-nearest neighbor algorithm classifies each piece of the test dataset according to the following steps. First, the space between the test instance and each training piece requires a calculation. Second, the k-nearest neighbor algorithm will sort all the distances according to the increasing correspondence of distance. Third, *k* points with the shortest distance are chosen. Then, the frequency of the *k* points in the category is identified. Finally, the class with the largest frequency is returned as the predictive classification of the test data[31].

3.1.2. Key steps of k-nearest neighbor algorithm

The three critical steps in the k-nearest neighbor algorithm are the selection of the *k* value, the calculation of distance, and the formulation of the classification strategy.

A small *k* value denotes that the training data, which is almost equal to the input data, plays a role in the last estimation category, but the overfitting issue could happen easily. While the value of *k* is enormous, the advantage is that the prediction error rate can be suppressed. But the drawback is that the approximate error increases[32] (Figure 3), and then the training instance which is far away from the test data will also affect the estimation, resulting in an error prediction. In practical application, a smaller value is generally chosen for the *k* value, and the best *k* value is usually chosen by the cross-validation method. When the number of training instances approaches infinity and $k = 1$, the error rate is at most twice the Bayesian error rate, and if the *k* value also approaches almost infinity, the error rate approaches the Bayesian error rate[33].

There are several ways to compute the space between the input sample and each training point. The most common methods are Euclidean and Manhattan[34]. The n-dimensional vectors *x* and *y* are represented separately from the data to be tested and a point in the sample set. The Euclidean distance calculates the square root of the sum of square variances between the input sample and the pieces in the training set:

$$\sqrt{\sum_{i=1}^{n}\left(x_i - y_i\right)^2} \quad \text{(I)}$$

The Manhattan distance is the sum of the absolute difference between the test sample and the points in the training set:

$$\sqrt{\sum_{i=1}^{n}\left|x_i - y_i\right|} \quad \text{(II)}$$

Before the calculation, the value of each feature should be normalized, which can help prevent the weights of

attributes with enormous initial values from being larger than those with smaller initial values.

The classification judgment rules in the algorithm are often majority voting[35]. The most class value of the k nearest training pieces of the input data determines its category. In the regression task, the average value of k nearest neighbors can be used as the predictive value.

3.2. Artificial neural network

Compared with traditional machine learning algorithms such as the k-nearest neighbor, a neural network has better generalization ability, and it can automatically extract the characteristics of sample data for classification, regression, and nonlinear operations. The simplest artificial neural network consists of three layers, an input layer, a hidden layer, and an output layer. The data comes in from the input layer, and after being processed by the hidden layer, the output layer outputs the final result[36]. The most critical section is the hidden layer, where most of the data feature extraction and generalization work is done. An artificial neural network containing multiple hidden layers is often called a deep neural network, and the corresponding learning process is called deep learning[37].

The data sequence from the input layer through the hidden layer and finally output in the output layer is considered to be positive. Such a neural network model is called a feedforward neural network. In the process of feedforward neural network model training, the parameters of each neuron are constantly modified until the final model can better fit the data in the training set. Feedforward neural network usually uses the back propagation algorithm based on gradient descent to update neuron parameters[38].

3.2.1. Feedforward neural network

The feedforward neural network is a classical neural network model, which is also called multilayer perceptron. Feedforward refers to the one-way propagation of parameters from the input end to the output end. The model itself and the output of the model do not form a directed ring[39]. There is no feedback connection, and the information always flows to the output end. When there is a feedback connection in the network, the model does not belong to the feedforward neural networks. In the forward propagation stage, the input feature vector is processed, and the output value of each node is calculated. If there is a deviation between the real value of the output layer and the expected value, the error is propagated back[40].

3.2.2. Back propagation algorithm

A back propagation refers to the order in which neurons update parameters first from the output layer, then from the hidden layer, and finally from the input layer, which is the opposite direction of the data propagation in the neural network model. The essence of neural network training is to adjust the parameters of each layer iteratively according to the error between the actual output and the expected output[41]. In the training stage, the gradient of the loss function is calculated layer by layer through the back propagation algorithm, the output error of each layer in the network is fed back to the upper layer through the iterative method, and then the weight of the network is updated layer by layer, so that the theoretical value is closer to the sample value. After several iterations, the output error of the objective function is reduced to the minimum value, and the trained model is obtained[42].

Gradient refers to the expression that expresses the partial derivative of a parameter in a function of multiple variables in the form of a vector. In the geometric sense, it represents the place where the function changes and increases the most. The gradient descent algorithm is a search-based optimization method, which is a common method to minimize the loss function. Its mathematical theory is the chain derivative rule of compound function. The gradient descent algorithm mainly obtains the gradient of the objective function for all variables in the process of training the neural network. In the operation of the neural network gradient descent, the weight is updated through the negative gradient direction, and the effective optimal way of the objective function can be obtained[43].

There are two commonly used gradient descent algorithms: batch gradient descent algorithm and random gradient descent algorithm. The difference between them mainly lies in the number of samples used to obtain gradient and update parameters. The former uses all samples, while the latter only selects one sample at random. Random gradient descent algorithm has advantages in training speed because only one sample is randomly selected to update parameters, while the batch gradient descent algorithm has more advantages in convergence speed and can converge to the local optimal point faster[44].

3.3. Convolutional neural network

The simple network structure of the artificial neural network model is the reason for the loss of spatial information in vector space, the difficulty of multi-parameter training, and the problems of network overfitting. However, a convolutional neural network can better solve the defects of artificial neural networks. Its main characteristics are local connection and parameter sharing, and it is easier to optimize the network by reducing the number of weights, thus reducing the risk of model overfitting. Convolutional neural networks have a significant improvement in large image processing performance compared with artificial neural networks[45].

The convolutional neural network is a kind of feedforward neural network that contains convolution and pooling computation as well as a depth structure. It is one of the classical algorithms of deep learning. The convolutional neural network framework is generally composed of the input layer, convolutional layer, pooling layer, fully connected layer, and output layer[46]. The depth of the network depends on the number of the convolutional layer, pooling layer, and fully connected layers, in which the order of pooling layer and convolutional layer can be changed. In the traditional neural network, the connections between neurons at each layer are fully connected, while in the convolutional neural network, the connections between neurons on each feature map are only connected with neurons in a small region of the upper layer. The hidden layer is alternately composed of convolutional layer and pooling layer. Features are extracted through convolution operation, and then more abstract features are obtained through pooling operation. Finally, the obtained feature map is input to the fully connected layer, and the result of the last fully connected layer operation is input to the output layer[47].

3.3.1. Convolution layer

The convolutional layer is a basic component of the convolutional neural network architecture. It mainly performs shallow feature extraction on the data transmitted to the network, such as image edge, texture, shape, and other features. Feature extraction usually refers to the combination of linear and nonlinear calculations, *i.e.*, convolution operation and activation function. The convolutional layer has two important properties. The neurons between the convolutional layers are connected to the local receptive field of the previous layer employing local connection and weight sharing. Compared with the fully connected network, the local connection mode can greatly reduce the number of network training parameters and speed up the training speed. Local connection means that neurons are only associated with a small number of pixels in the input image. For image data, the correlation between adjacent pixels is greater than that between pixels far apart, *i.e.*, the local correlation of the image. A local connection aims to extract the local features of the image by using this feature and combine it into the global features of the image at the deep level of the network. Although the network using local connection will reduce the parameters, the order of magnitude of the parameters is still large, so the concept of weight sharing is proposed to further reduce the network training parameters[48].

3.3.2. Pooling layer

The pooling layer is associated with the convolutional layer, and the feature maps output by the convolutional layer needs to be processed by pooling. The pooling function will carry out statistical selection and information filtering on the input features to adjust the output data. Pooling layers select the pooling area for pooling operations. Common operations include mean pooling, maximum pooling, and mixed pooling. The use of pooling reduces the size of the feature data, as well as the size of the input data in the next layer, improves the efficiency of data statistics, and reduces the quantity storage space[49].

3.3.3. Fully connected layer

Fully connected layers in convolutional neural networks act as "classifiers," which can map the learned features and distributed representations to the label space. It can be simply understood as combining the features extracted from the previous layers into a single output value, which can reduce the influence of feature location on classification. Convolutional neural networks connect the data to one or more fully connected layers after passing through several convolutional and pooling layers. Each neuron in a fully connected layer is fully connected to all neurons in the previous layer for output. The local information with class distinguishing features in the convolutional layer and the pooling layer will be integrated by the fully connected layer, and the output value of the last fully connected layer is the corresponding probability[50].

3.4. Long short-term memory

A long short-term memory network is a modified recurrent neural network, which can remember long- and short-term information. It can not only deal with the long distance dependence problem that the recurrent neural network cannot manage, but also solve common issues such as gradient explosion or gradient disappearance in the neural network[51]. Therefore, it is very outstanding in handling sequence data. Long short-term memory networks are suitable for treating and evaluating critical information with long distance and delays in time series. A long short-term memory network is a variant of the recurrent neural network, whose core concepts are cell state and gate mechanism[52] (Figure 4).

3.4.1. Cell state

The cell state corresponds to the way information is being conveyed, so that information can be transported in the sequence. It can be regarded as the memory of the network. In theory, the cell state can transfer the related knowledge during the sequence handling. As a result, the information in the earlier time stage can even be carried into the cells during the later time stage, which conquers the impact of short-term memory[53].

The cell state of the previous layer is multiplied by the forgetting vector point by point. If it is multiplied by a

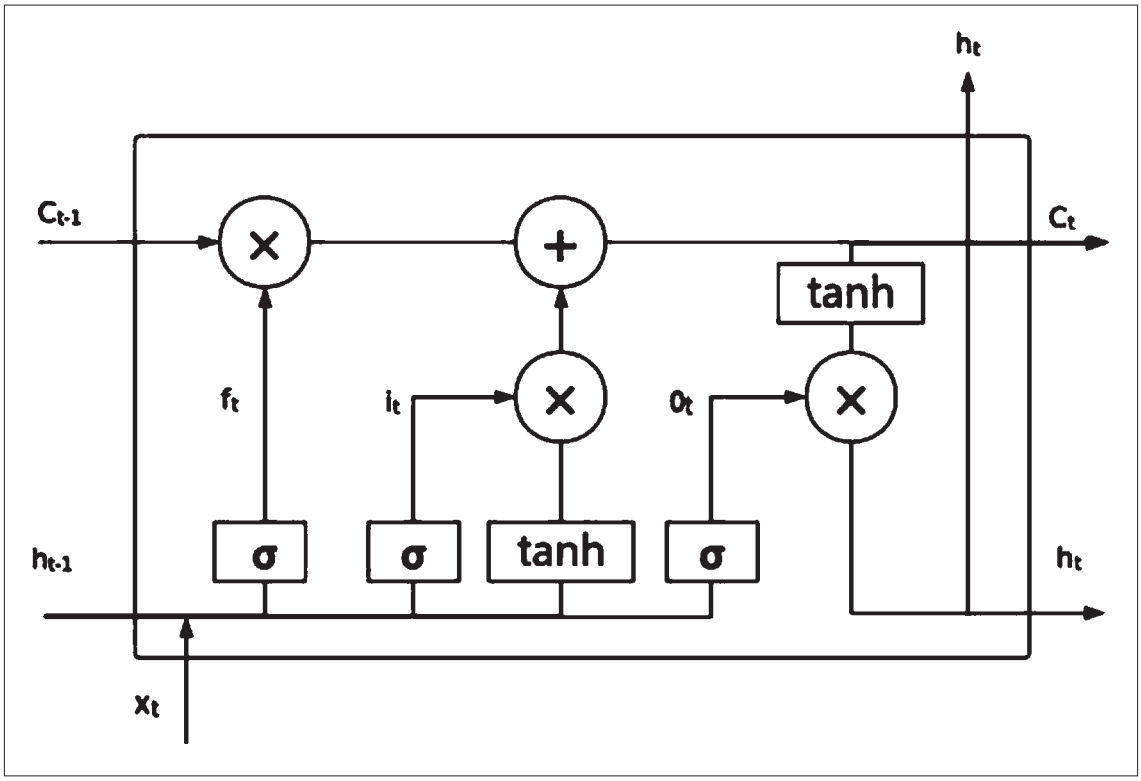

Figure 4. Structure of the long short-term memory algorithm. Reprinted from ref. [52] under the terms of the Creative Commons CC-BY license.

number that is almost zero, this information needs to be abandoned in the new cell state. Then, the value is added point by point with the output result of the input gate to update the latest information determined by the neural network to the cell state. Thus, the updated cell state is obtained[54].

3.4.2. Gate mechanism

The reason that long short-term memory can deal with the long-term dependence issue of the recurrent neural network is that long short-term memory integrates a gate mechanism to control the flow and loss of attributes[55]. The addition and removal of information are controlled by a gate mechanism that learns what information should be saved or forgotten during the training procedure[56].

Forgetting gate. After new information is input, if the architecture will forget the old information, the forgetting gate is used to accomplish it. The forgetting gate is a vital element of the long short-term memory network section, which can control what information to retain and what information to forget, and in some way avoid the issue of gradient vanishing and gradient explosion caused by the reverse propagation of gradient over time. The forgetting gate identifies what information the long short-term memory networks discard from the cellular state of the previous moment. The gate reads the relevant information and maps it via a function to a value between zero and one, which is then multiplied by the cell state to determine what information to discard. When its value is one, the information is completely kept, and when the value is zero, the information is completely discarded[57].

Input gate. Input gate is utilized to revise cell state. First, the information about the earlier hidden status and the existing input is converted to the *sigmoid* function. The value will be adjusted into the range of zero to one to determine what information to modify. Zero means not essential, and one means crucial. Second, the information of the preceding hidden state and the current input information is transmitted to the *tanh* function to obtain a new candidate value vector. Finally, the output value of the *sigmoid* function is multiplied by the output value of the *tanh* function. The output result of the *sigmoid* function will identify which information in the output result of the *tanh* function is significant and needs to be retained[58,59].

Output gate. Output gate is employed to predict the output of the subsequent hidden state, which includes the information that has been input earlier. First, we pass the previous hidden state and the current input to the *sigmoid* function, and then deliver the updated cell state to the *tanh* function. Finally, the result of the *tanh* function is multiplied by the production of the *sigmoid* function

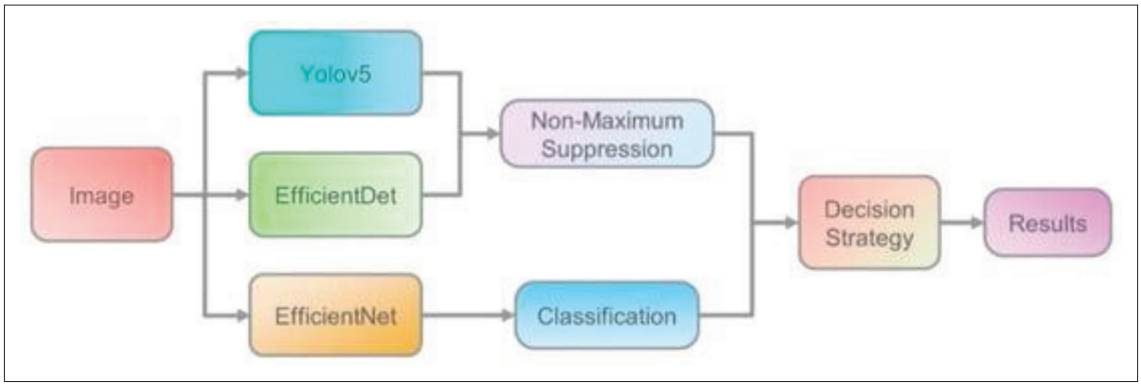

Figure 5. An ensemble learning framework. Reprinted from ref. [62] under the terms of the Creative Commons CC-BY license.

to get the information the hidden state should own. The hidden status is then taken as the result of the current cell, and the latest cell state and the refreshed hidden state are transferred to the subsequent time stage[60].

3.5. Ensemble learning

Ensemble learning is the procedure of training several base machine learning frameworks and combining their results[61]. Based on different models, the organization is committed to building an excellent prediction framework. The combination of the machine learning frameworks can enhance the stability of the overall scheme and acquire more precise estimation results. Ensemble learning models are usually more reliable than individual models[62] (Figure 5).

3.5.1. Combination model of ensemble learning

Ensemble learning combines several weak classifiers through a specific strategy to produce a robust classifier, and the classification precision of the strong classifier is much better than each weak classifier[63]. The combination strategies of ensemble learning are divided into three types:

Bagging. Bagging methods improve the independence of each base model by randomly constructing training samples and selecting features. Due to the different training data, the obtained learner will be different. However, if each subset of the sample is entirely different, each base learner can only train a small part of the data and cannot carry out effective learning. Therefore, sampling subsets are often obtained by overlapping each other[64].

Boosting. The essential difference between boosting and bagging is the diverse way they treat the base models. The boosting method selects the elite through continuous testing and screening, and then gives the elite more voting rights[65]. The poor performance of the basic model provides fewer voting right, and then synthesizes the voting of all models to get the final result.

Stacking. There are two main differences between bagging, boosting, and stacking. First, stacking generally regards heterogeneous vulnerable learners, where various learning algorithms are combined. However, bagging and boosting commonly regard homogeneous weak learners. Secondly, the stacking law habitually connects the basic model with the met model, while bagging and boosting incorporate the weak learner based on the deterministic algorithm[66].

3.5.2. Result generation of ensemble learning

For different prediction results that multiple weak classifiers produce, ensemble learning usually uses the following three methods to make the final result.

Average method. For the regression evaluation problem of numerical classes, the typically applied integration strategy is averaging. It averages the outputs of several weak classifiers to calculate the last result. The most straightforward average is the arithmetic average. In addition, we can multiply each result by a weight to calculate weighted average[67,68].

Voting method. The simplest voting algorithm is the relative majority voting method, and the result that appears most often is the last category. If several categories obtain the highest vote simultaneously, one is randomly chosen as the final result. The slightly more complex voting solution is the absolute majority voting way, *i.e.*, more than half of the votes. Based on the relative majority voting algorithm, not only the highest number of votes is needed, but the highest number of votes should also exceed half at the same time. Otherwise, the estimation will be rejected[69]. A more sophisticated approach is the weighted voting way. Similar to the weighted average method, the number of classified votes for each weak learner is multiplied by a weight, and the weighted votes of each type are finally summed. The class corresponding to the maximum value is the final result.

Learning method. The last two methods average or vote on the outputs of weak learners, which are sufficiently straightforward. But they may have a significant learning

error, and the learning method is designed. The learning method is an easy logical processing of a layer of learners, instead of the results of weak learners. The results of training weak learners are input, and a learner is retrained to obtain the last output. In that case, we call the weak learner the primary learner, and the learner used for combination is called the secondary learner. For the test sample, the primary learner is first predicted to gain the input data of the secondary learner. Then, the secondary learner is predicted again to obtain the final prediction result[70].

3.6. Comparison of the machine learning methods

The traditional machine learning algorithm represented by K-nearest neighbor is generally considered shallow learning, and the features of the training data must be specified manually in the training process. Its feature extraction ability of sample data and model generalization is relatively weak, and it is usually applied in classification and linear regression problems. Machine learning algorithms represented by neural networks are considered deep learning, which can automatically extract features from training data without human intervention. Therefore, deep learning has better feature extraction ability and model generalization ability as well as good analysis effect on high-dimensional data, and can deal with nonlinear problems. However, the deep learning model has poor explanatory ability, involves many parameters, and consumes a lot of resources in the training process. However, the traditional machine learning algorithm has a complete mathematical theory and strong interpretability. Moreover, the training of traditional machine learning algorithm models require fewer resources[71].

Multi-layer perceptron, convolutional neural networks, and long short-term memory neural networks all belong to deep learning. When there are too many layers in the multi-layer perceptron model, there will be too many model parameters, resulting in high training costs and overfitting problems. Convolutional neural network solves these problems by using local connection and parameter sharing to some extent. However, the data in the convolutional neural network can only travel forward and have no memory of the data that have been processed. Long short-term memory neural network not only has memory function, but also can deal with long-term dependence problem. But the structure of long short-term memory neural network is too complex[72].

Regardless of how a single machine learning model is trained and optimized, it always has some shortcomings that are difficult to solve. Ensemble learning can combine a variety of machine learning algorithms to get a better model to obtain better predictive performance. The integrated learning algorithm itself is not a single machine learning algorithm, but through the construction and combination of multiple machine learning machines to complete the learning task, it has a high accuracy in the machine learning algorithm, with the shortcomings being the complicated training process of the scheme and the low efficiency[67].

4. Looking for more suitable bioinks

Bioprinted products are composed of bioinks that are deposited layer by layer, and therefore, the performance of bioinks has a significant impact on the final bioprinted products. Due to the variety of bioprinting technologies, it is difficult to find a unified standard to prepare bioink. Advanced technology and high level of operators' experience are often required to manually prepare bioink, and the quality of final product is not guaranteed. The machine learning algorithm, combined with its powerful feature analysis ability, gives a perfect solution for the preparation of bioink through the configuration of bioink in the existing cases and addressing the inconsistent quality of the final printed product.

The cell activity in bioink has a very important influence on the final printing effect. However, the low activity of cells in extrusion-based bioprinting limits its application. Reina-Romo *et al.*[73] attempted to quantitatively analyze the effects of cells in bioink based on machine learning algorithms. They studied the force of three hydrogels when passing through two nozzles of different shapes. The collected data were analyzed by a machine learning algorithm called Gaussian process. The algorithm was not only suitable for estimating the importance of each parameter, but also could calculate the influence of changing parameters on the final printing results. The dataset was derived from the existing accurate data, and its adaptability was verified by random partition. The effect of the framework was verified by comparing the calculated results with the current experimental data. In order to maintain cell activity and withstand the pressure in the printing process, Allencherry *et al.*[74] studied the configure method of hydrogel and gelatin bath in extrusion-based bioprinting. A convolutional neural network was employed to identify the quality of the final prints. The objects processed by the neural network and the data in the dataset were all images obtained by optical microscope scanning, with a total of 108 images in the dataset. A statistical analysis reported that the accuracy of the model could reach 93.51%. Xu *et al.*[75] introduced a framework containing four machine learning algorithms to determine viability of cells in bioink in stereolithography-based bioprinting. The four algorithms are random forest, k-nearest neighbor, ridge regression, and neural networks. They first continuously

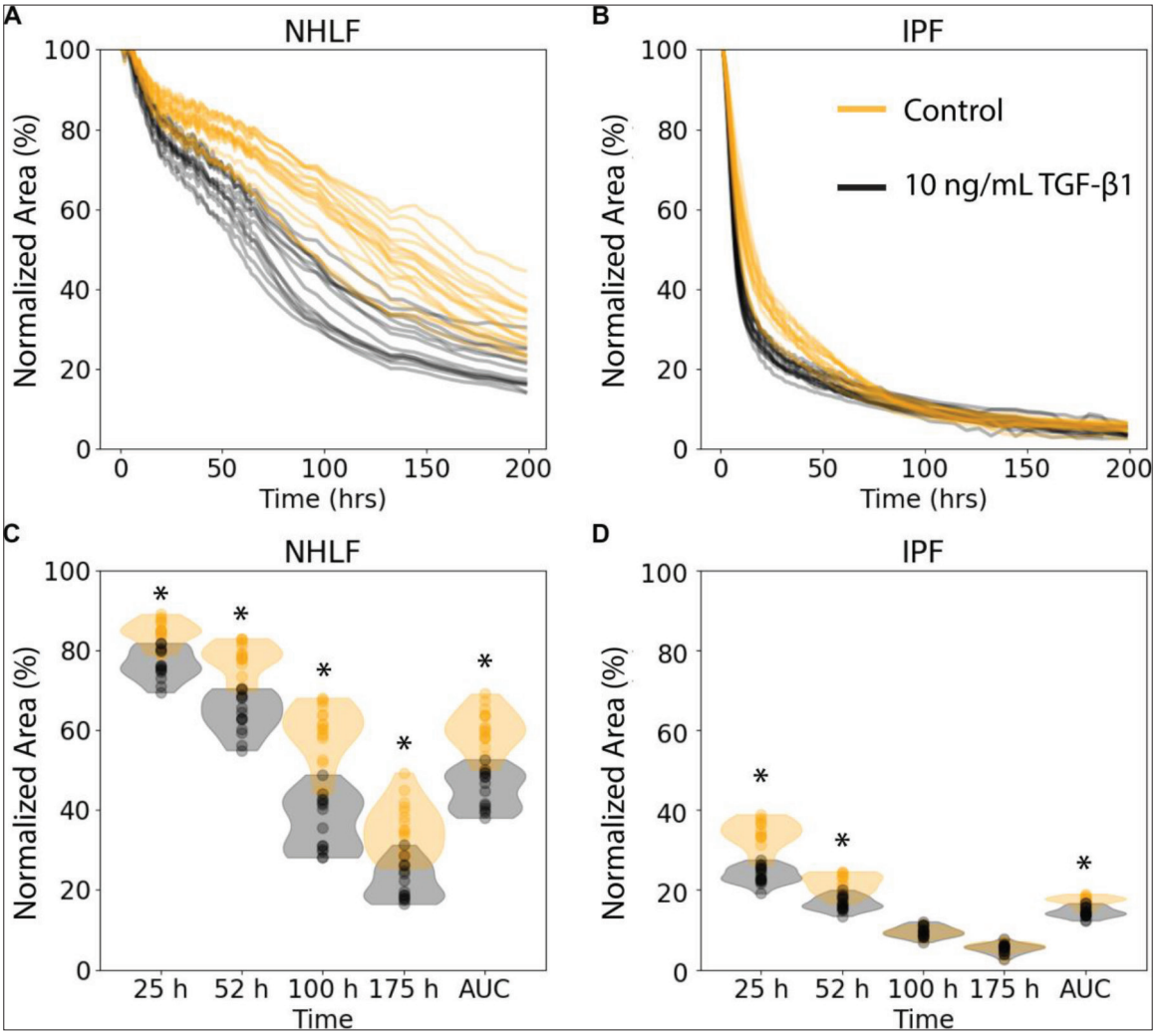

Figure 6. Shrinkage behavior of collagen microgel bioink. (A) The normalized region of normal human lung fibroblast (NHLF) bioink. (B) The normalized region of idiopathic fibrosis (IPF) bioink. (C) Response of NHLF to stimulation in bioink. (D) Response of IPF cells to stimulation in bioink. Reprinted from ref.[78] under the terms of the Creative Commons CC-BY license.

changed the experimental parameters to test the viability of cells in the bioink to obtain the result data. These data were randomly divided into a training set and a validation set and used to train machine learning algorithms. Based on the framework, they validated the influence mechanism of ultraviolet intensity, exposure time, and other parameters on cell activity in bioink.

Based on ensemble learning algorithms, Wu *et al.*[76] predicted the speed and volume of droplets formed by bioink during inkjet bioprinting. Their ensemble learning system included machine learning algorithms such as support vector machine, regularized least squares regression, and random forest. The dataset was derived from images of the droplet formation process taken by the imaging system. The experimental data confirmed that the model could accurately predict the speed and volume of droplets formed by bioink. In order to obtain bioprinted products that meet both mechanical properties and biocompatibility, Lee *et al.*[77] studied the method of generating bioinks through machine learning algorithms. They first analyzed the relationship between rheology and printability using the relative minimum general generalization algorithm. Then, they mined the results based on the multiple regression algorithm to obtain the formulation of bioink. They determined that high elastic modulus could improve shape fidelity, and it could be extruded below the critical yield stress. Subsequently, they used hydrogels to generate a 3D-bioprinted product and confirmed the conclusion.

By 3D-bioprinting collagen matrix materials, Yamanishi *et al.*[78] simulated the process of pulmonary fibrosis *in vitro* (Figure 6). They employed machine

learning algorithms to analyze the shrinkage kinetics of bioinks used in the printing process. The dataset contained 10,000 photos taken by a microscope in the incubator. The machine learning algorithm was the knowledge analysis segmentation, which had been coupled with off-the-shelf software and could directly process images. The team finally determined the reaction curve of bioink inhibited by three small molecules. Tourlomousis *et al.*[79] proposed a method of controlling cell phenotype by regulating the biophysical properties of cells in bioink. All the cell images applied in the study were obtained by a 3D confocal microscope, and then multi-dimensional features of cells were extracted from the pictures. The dataset was also constructed by multi-dimensional features of cells extracted from images. The machine learning algorithm for classifying cells after adjusting biophysical characteristics was the support vector machine algorithm. The research results have important reference significance for the design of materials for single cell bioprinting.

Safir *et al.*[80] utilized blood as an example to investigate the detection method of bacteria in bioink in acoustic printing. They divided the bioink into many droplets, each containing only one or several cells. Then, they analyzed each droplet by Raman spectroscopy and obtained a large amount of experimental data. These data were objects processed by machine learning algorithms and components of the dataset used to train machine learning algorithms. The results of the random forest algorithm for bacteria classification were compared with the results of scanning electron microscopy images, which proved the effectiveness of the framework. The framework had a high accuracy of 99% in a single bacterial droplet. Even in mixed bacterial droplets, it could still reach 87%. Due to the lack of printability, bioinks made of hydrogels often failed to perform 3D bioprinting. However, due to the low cost of hydrogels, the research community has been looking for hydrogel formulations that can be used as bioinks. Nadernezhad *et al.*[81] analyzed the data based on machine learning algorithms to reveal the recipe for transforming hydrogels into printable bioinks. The fundamental experimental data were processed by MATLAB software and random forest algorithm. The dataset of the training algorithm was composed of a random selection of data. Eventually, the researchers identified 13 indicators that had a crucial influence on the bioprinted products, which had a positive effect on the formulation of hydrogels which were transformed into bioinks.

5. Parameter optimization of 3D bioprinting

Traditionally, many trial-and-error experiments are required to find the appropriate 3D bioprinting parameters. When a set of printing parameters do not achieve the desired results, it is necessary to constantly change the reference effect until a satisfactory result is obtained. This method not only consumes a lot of time and energy, but also causes a lot of waste of raw materials. Even if good results have been achieved, it is difficult to apply the parameters in combination to other bioprinting activities due to the differences in bioinks and 3D bioprinting methods. The powerful learning ability of the machine learning algorithm based on the existing data can yield a better combination of printing parameters for other bioprinting cases after the model is trained. The operation can save a lot of time and cost, and the versatility is better than the original method.

The bioink used in 3D bioprinting based on digital light processing has a light scattering effect, which will affect the final printing effect. Traditionally, the printing parameters need to be optimized to compensate for the scattering effect. However, this method is laborious and will cause serious waste of printing materials. By learning the existing optimization case data through a deep neural network, Guan *et al.*[82] developed a parameter optimization method that can automatically compensate for the light scattering effect of printing bioink. The dataset for training deep neural networks was composed of 4,000 pairs of data generated by program simulation. Experiments showed that after optimizing the printing parameters by the method, the 3D bioprinter had an excellent compensation effect on the light scattering effect of bioink. Bone *et al.*[83] proposed a hierarchical machine learning framework to determine the optimal parameters for 3D bioprinting based on alginate hydrogel, involving printing speed, bioink concentration, nozzle diameter, *etc.* The framework was divided into three layers, and the support vector machine algorithm was introduced. The dataset was obtained by comparing the final product with the design model by changing the printing parameters. This approach was particularly suitable for parameter prediction in the bioprinting of complex structures. Shi *et al.*[84] conducted multi-parameter optimization of inkjet bioprinting based on a fully connected neural network. The fully connected neural network included four layers: an input layer, two hidden layers, and an output layer. Multiple results generated by the simulation program formed the dataset for training the fully connected neural network, containing 120 data. Experiments revealed that the inkjet bioprinting parameters optimized by the method could significantly promote the precision and stability of printing. In order to enhance the accuracy of drop-on-demand printing, the team utilized a multi-layer perceptron neural network to find suitable printing parameters[85]. The architecture coupled computational fluid dynamics model and neural network algorithm, and optimized parameters such as bioink viscosity and nozzle diameter based on classification.

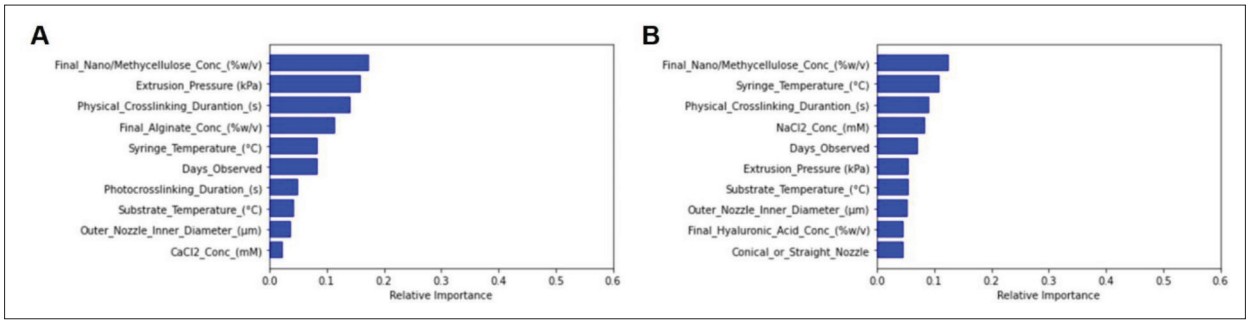

Figure 7. (A) Feature weight in random forest regression and (B) feature weight in random forest classification. Reprinted from ref. [87] under the terms of the Creative Commons CC-BY license.

The dataset was composed of 99 simulated data obtained by the simulation program. The results indicated that the neural network with four hidden nodes in a hidden layer had the best prediction accuracy, which could reach 90%.

Fu *et al.*[86] studied the optimization method of parameters in extrusion-based bioprinting based on a support vector machine algorithm. The parameters involved path height, nozzle temperature, nozzle size, and composition. In addition to the optimization method for determining the printing parameters, the researchers also investigated the training effect of small-scale data sets on machine learning algorithms. There were only 12 pieces of data in the dataset they used, which were obtained in actual experiments. Using the support vector machine model created by them to evaluate the parameters in advance before extrusion-based bioprinting could ensure that the probability of high-quality printing was higher than 75%. In addition, they also studied the universality of the model. Tian *et al.*[87] collected data from published literature and trained the machine learning regression model to find better parameters in extrusion-based bioprinting (Figure 7). The bioink contained alginate and gelatin, and the studied parameters included cell viability and extrusion pressure. Support vector regression and random forest regression were utilized in the regression model, and the dataset contained 956 data. The calculation results verified that the model could not only predict the printing effect in the original literature after a lot of training, but also further analyze the details. The research results once again confirmed the effectiveness of machine learning in the design of bioprinting experiments. Ruberu *et al.*[88] investigated the printability of extrusion-based bioprinting based on the Bayesian optimization algorithm in a search for better printing parameters. Unlike other machine learning algorithms, the Bayesian optimization algorithm did not need a lot of data to train the model. Its advantage was that the construction of the architecture was completed with only a few experimental data. The experimental results showed that compared with the previous methods, the model could also optimize the printing parameters while significantly reducing the number of experiments.

6. Defect detection during 3D bioprinting

It is inevitable to have various problems in the 3D bioprinting process, which may lead to the final product not meeting the relevant requirements. If the manual method is used to monitor errors in the printing procedure, the method would be time-consuming and labor-intensive, and require high level of skill and experience of the operator. The quality of 3D-bioprinted products is difficult to guarantee. Machine learning algorithms could automatically monitor the printing process and detect defects in time by automatically learning the characteristics of existing defective products. Therefore, machine learning algorithms are also widely applied in the defect detection of 3D bioprinting.

Jin *et al.*[89] designed four models based on machine learning algorithms to detect anomalies on each layer during 3D bioprinting. Support vector machine and deep neural network algorithms are employed in the four models. A dataset including 240 images was constructed by the team to detect defects in the transparent bioprinting process and optimize the parameters of the printing process. Experiment results revealed that the model with a conventional neural network worked best. They expected that introducing transfer learning into the model would help reduce the burden of building a dataset and improve its versatility. In order to find out the possible quality problems in the extrusion-based bioprinting process in time, Bonatti *et al.*[90] added a long short-term memory neural network to the bioprinting control process (Figure 8). A camera was placed in front of the printer to film the printing process and extract the dataset for training the neural network. The team constantly modified the printing parameters to obtain videos of the bioprinting in different states, which ultimately improved the comprehensiveness of the dataset. Experimental results showed that the method could

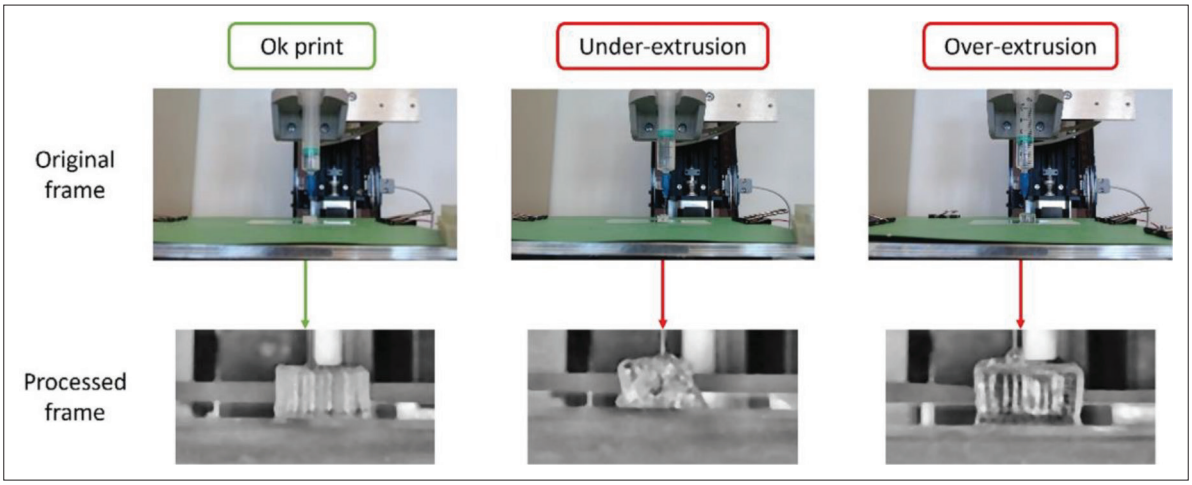

Figure 8. Image pre-processing during defect detection of extrusion-based bioprinting. Reprinted from ref. [90] under the terms of the Creative Commons CC-BY license.

effectively detect quality errors during 3D bioprinting, and a bigger error signified that a higher detection accuracy was reached. Tebon *et al.*[91] utilized a conventional neural network to detect high-throughput drug screening problems in 3D-bioprinted organs. A quantitative phase imaging technology was employed to obtain relevant images. The training dataset contained 100 manually marked organ-like objects in randomly selected imaging frames. The team demonstrated that the detection effect of the method was good, providing a new solution for the rapid screening of 3D-bioprinted organs. Tröndle *et al.*[92] obtained a renal sphere by bioprinting and tested its toxicity based on deep learning technology. Due to the precise deposition of low-volume, low-viscosity bioink, the renal sphere was generated by drop-on-demand bioprinting. They obtained the relevant images by fluorescently labeling toxic substances on the kidney sphere. A hyper-parameter Bayesian-optimized convolutional neural network evaluated the toxicity of renal spheres through image processing. The dataset utilized to train the neural network was created by the researchers using single cell images. Experiments showed that the accuracy of the model could reach 78.7%.

7. Conclusion

The rapid development of 3D bioprinting technology in recent years is accompanied by the outstanding outcomes regarding its application. Although there are still many problems to overcome, new attempts have been made. In recent years, machine learning has been applied in many cases, which gives a solid impetus of incorporating machine learning to various fields, and the scope of application is also expanding. This paper summarizes the research developments of machine learning in the field of 3D bioprinting in recent years, hoping to promote the combination of the two technologies further. First, the basic principles of k-nearest neighbor, long short-term memory, and ensemble learning are introduced. Then, the application of machine learning in additive manufacturing is reviewed. The in-depth analysis of additive manufacturing technology is helpful to study 3D bioprinting, which is essentially an additive manufacturing technology. Finally, the existing work of machine learning in bioink preparation, printing parameter optimization, and printing defect detection of 3D bioprinting are summarized. It is expected that this paper can inspire more and better research and development of methods concerning the combination of 3D bioprinting and machine learning.

Acknowledgments

None.

Funding

This work is funded by the National Research Foundation of Korea (No. NRF-2022R1A2C2002799).

Conflict of interest

The authors declare no conflict of interest.

Author contributions

Conceptualization: Sang Woo Joo
Writing – original draft: Hongwei Ning
Writing – review & editing: Teng Zhou, Sang Woo Joo

Ethics approval and consent to participate

Not applicable.

Consent for publication

Not applicable.

Availability of data

Not applicable.

References

1. Xiaoya Z, Liuchao J,Jingchao J, 2022, A survey of additive manufacturing reviews. *MSAM*, 1(4):21.
2. Jiang J, 2020, A novel fabrication strategy for additive manufacturing processes. *J Clean Prod*, 272:122916.
   https://doi.org/10.1016/j.jclepro.2020.122916
3. Al-Kharusi G, Dunne NJ, Little S, *et al.*, 2022, The role of machine learning and design of experiments in the advancement of biomaterial and tissue engineering research. *Bioengineering*, 9(10):561.
4. Gholami P, Ahmadi-pajouh MA, Abolftahi N, *et al.*, 2018, Segmentation and measurement of chronic wounds for bioprinting. *IEEE J Biomed Health*, 22(4):1269–1277.
   https://doi.org/10.1109/JBHI.2017.2743526
5. Shin J, Lee Y, Li Z, *et al.*, 2022, Optimized 3D bioprinting technology based on machine learning: A review of recent trends and advances. *Micromachines*, 13(3):363.
6. Ravanbakhsh H, Karamzadeh V, Bao G, *et al.*, 2021, Emerging technologies in multi-material bioprinting. *Adv Mater*, 33(49):2104730.
   https://doi.org/10.1002/adma.202104730
7. Malekpour A, Chen X, 2022, Printability and cell viability in extrusion-based bioprinting from experimental, computational, and machine learning views. *J Funct Biomater*, 13(2):40.
8. Ong CS, Yesantharao P, Huang CY, *et al.*, 2018, 3D bioprinting using stem cells. *Pediatr Res*, 83(1):223–231.
   https://doi.org/10.1038/pr.2017.252
9. Sklare SC, Richey WL, Vinson BT, *et al.*, 2017, Directed self-assembly software for single cell deposition. *Int J Bioprint*, 3(2):100–108.
10. Datta P, Barui A, Wu Y, *et al.*, 2018, Essential steps in bioprinting: From pre- to post-bioprinting. *Biotechnol Adv*, 36(5):1481–1504.
    https://doi.org/10.1016/j.biotechadv.2018.06.003
11. Zolfagharian A, Denk M, Kouzani AZ, *et al.*, 2020, Effects of topology optimization in multimaterial 3D bioprinting of soft actuators. *Int J Bioprint*, 6(2):50–60.
12. Ng WL, Chan A, Ong YS, *et al.*, 2020, Deep learning for fabrication and maturation of 3D bioprinted tissues and organs. *Virtual Phys Prototyp*, 15(3):340–358.
    https://doi.org/10.1080/17452759.2020.1771741
13. He H, Yang Y, Pan Y, 2019, Machine learning for continuous liquid interface production: Printing speed modelling. *J Manuf Syst*, 50:236–246.
    https://doi.org/10.1016/j.jmsy.2019.01.004
14. Yu C, Jiang J, 2020, A perspective on using machine learning in 3D bioprinting. *Int J Bioprint*, 6(1):1–8.
15. Freeman S, Calabro S, Williams R, *et al.*, 2022, Bioink formulation and machine learning-empowered bioprinting optimization. *Front Bioeng Biotechnol*, 10:913579.
16. Jin Z, Zhang Z, Demir K, *et al.*, 2020, Machine learning for advanced additive manufacturing. *Matter*, 3(5):1541–1556.
    https://doi.org/10.1016/j.matt.2020.08.023
17. Omairi A, Ismail Z H, 2021, Towards machine learning for error compensation in additive manufacturing[J]. *Appl Sci*, 11(5):2375.
    https://doi.org/10.3390/app11052375
18. Qin J, Hu F, Liu Y, *et al.*, 2022, Research and application of machine learning for additive manufacturing. *Addit Manuf*, 52:102691.
    https://doi.org/10.1016/j.addma.2022.102691
19. Jiang J, Xu X, Xiong Y, *et al.*, 2020, A novel strategy for multi-part production in additive manufacturing. *Int J Adv Manuf Technol*, 109(5):1237–1248.
    https://doi.org/10.1007/s00170-020-05734-8
20. Ko H, Witherell P, Lu Y, *et al.*, 2021, Machine learning and knowledge graph based design rule construction for additive manufacturing. *Addit Manuf*, 37:101620.
    https://doi.org/10.1016/j.addma.2020.101620
21. Li Z, Zhang Z, Shi J, *et al.*, 2019, Prediction of surface roughness in extrusion-based additive manufacturing with machine learning. *Robot Cim-Int Manuf*, 57:488–495.
    https://doi.org/10.1016/j.rcim.2019.01.004
22. Zhu Z, Anwer N, Huang Q, *et al.*, 2018, Machine learning in tolerancing for additive manufacturing. *CIRP Ann*, 67(1):157–160.
    https://doi.org/10.1016/j.cirp.2018.04.119
23. Jiang J, Xiong Y, Zhang Z, *et al.*, 2022, Machine learning integrated design for additive manufacturing. *J Intell Manuf*, 33(4):1073–1086.
    https://doi.org/10.1007/s10845-020-01715-6

24. Ghayoomi Mohammadi M, Mahmoud D, Elbestawi M, 2021, On the application of machine learning for defect detection in L-PBF additive manufacturing. *Opt Laser Technol*, 143:107338.

    https://doi.org/10.1016/j.optlastec.2021.107338

25. Caggiano A, Zhang J, Alfieri V, *et al.*, 2019, Machine learning-based image processing for on-line defect recognition in additive manufacturing. *CIRP Ann*, 68(1):451–454.

    https://doi.org/10.1016/j.cirp.2019.03.021

26. Gobert C, Reutzel EW, Petrich J, *et al.*, 2018, Application of supervised machine learning for defect detection during metallic powder bed fusion additive manufacturing using high resolution imaging. *Addit Manuf*, 21:517–528.

    https://doi.org/10.1016/j.addma.2018.04.005

27. Li R, Jin M, Paquit VC, 2021, Geometrical defect detection for additive manufacturing with machine learning models. *Mater Design*, 206:109726.

    https://doi.org/10.1016/j.matdes.2021.109726

28. Bezdek JC, Chuah SK, Leep D, 1986, Generalized k-nearest neighbor rules. *Fuzzy Set Syst*, 18(3):237–256.

    https://doi.org/10.1016/0165-0114(86)90004-7

29. Gou J, Ma H, Ou W, *et al.*, 2019, A generalized mean distance-based k-nearest neighbor classifier. *Expert Syst Appl*, 115:356–372.

    https://doi.org/10.1016/j.eswa.2018.08.021

30. Wang C, Shi Y, Fan X, *et al.*, 2019, Attribute reduction based on k-nearest neighborhood rough sets. *Int J Approx Reason*, 106:18–31.

    https://doi.org/10.1016/j.ijar.2018.12.013

31. Gallego A-J, Calvo-Zaragoza J, Valero-Mas JJ, *et al.*, 2018, Clustering-based k-nearest neighbor classification for large-scale data with neural codes representation. *Pattern Recogn*, 74:531–543.

    https://doi.org/10.1016/j.patcog.2017.09.038

32. Zhang Q, Zhou H, Jiang Y, *et al.*, 2019, A simple joint modulation format identification and OSNR monitoring scheme for IMDD OOFDM transceivers using K-nearest neighbor algorithm. *Appl Sci*, 9(18):3892.

33. Ertuğrul ÖF, Tağluk ME, 2017, A novel version of k nearest neighbor: Dependent nearest neighbor. *Appl Soft Comput*, 55:480–490.

    https://doi.org/10.1016/j.asoc.2017.02.020

34. Shahabi H, Shirzadi A, Ghaderi K, *et al.*, 2020, Flood detection and susceptibility mapping using sentinel-1 remote sensing data and a machine learning approach: Hybrid intelligence of bagging ensemble based on k-nearest neighbor classifier. *Remote Sens*, 12(2):266.

35. Zhang Z, Jiang T, Li S, *et al.*, 2018, Automated feature learning for nonlinear process monitoring—An approach using stacked denoising autoencoder and k-nearest neighbor rule. *J Process Contr*, 64:49–61.

    https://doi.org/10.1016/j.jprocont.2018.02.004

36. Basheer IA, Hajmeer M, 2000, Artificial neural networks: Fundamentals, computing, design, and application. *J Microbiol Methods*, 43(1):3–31.

    https://doi.org/10.1016/S0167-7012(00)00201-3

37. Wu Y-c, Feng J-w, 2018, Development and application of artificial neural network. *Wireless Pers Commun*, 102(2):1645–1656.

    https://doi.org/10.1007/s11277-017-5224-x

38. Huang Y, 2009, Advances in artificial neural networks–methodological development and application[J]. *Algo*, 2(3):973–1007.

    https://doi.org/10.3390/algor2030973

39. Yang GR, Wang X-J, 2020, Artificial neural networks for neuroscientists: A primer. *Neuron*, 107(6):1048–1070.

    https://doi.org/10.1016/j.neuron.2020.09.005

40. Abiodun OI, Jantan A, Omolara AE, *et al.*, 2018, State-of-the-art in artificial neural network applications: A survey. *Heliyon*, 4(11):e00938.

    https://doi.org/10.1016/j.heliyon.2018.e00938

41. Vlachas PR, Pathak J, Hunt BR, *et al.*, 2020, Backpropagation algorithms and reservoir computing in recurrent neural networks for the forecasting of complex spatiotemporal dynamics. *Neural Netw*, 126:191–217.

    https://doi.org/10.1016/j.neunet.2020.02.016

42. Chan LW, Fallside F, 1987, An adaptive training algorithm for back propagation networks. *Comput Speech Lang*, 2(3):205–218.

    https://doi.org/10.1016/0885-2308(87)90009-X

43. Hameed AA, Karlik B, Salman MS, 2016, Back-propagation algorithm with variable adaptive momentum. *Knowledge-Based Syst*, 114:79–87.

    https://doi.org/10.1016/j.knosys.2016.10.001

44. Wright LG, Onodera T, Stein MM, *et al.*, 2022, Deep physical neural networks trained with backpropagation. *Nature*, 601(7894):549–555.

    https://doi.org/10.1038/s41586-021-04223-6

45. Lawrence S, Giles CL, Ah Chung T, *et al.*, 1997, Face recognition: a convolutional neural-network approach. *IEEE Trans Neural Netw*, 8(1):98–113.

    https://doi.org/10.1109/72.554195

46. Jaderberg M, Simonyan K, Vedaldi A, *et al.*, 2016, Reading text in the wild with convolutional neural networks. *Int J Comput Vis*, 116(1):1–20.

    https://doi.org/10.1007/s11263-015-0823-z

47. Acharya UR, Oh SL, Hagiwara Y, *et al.*, 2017, A deep convolutional neural network model to classify heartbeats. *Comput Biol Med*, 89:389–396.

https://doi.org/10.1016/j.compbiomed.2017.08.022

48. Zhou D-X, 2020, Theory of deep convolutional neural networks: Downsampling. *Neural Netw*, 124:319–327.

https://doi.org/10.1016/j.neunet.2020.01.018

49. Yamashita R, Nishio M, Do RKG, *et al.*, 2018, Convolutional neural networks: An overview and application in radiology. *Insights Imaging*, 9(4):611–629.

https://doi.org/10.1007/s13244-018-0639-9

50. Brodrick PG, Davies AB, Asner GP, 2019, Uncovering ecological patterns with convolutional neural networks. *Trends Ecol Evol*, 34(8):734–745.

https://doi.org/10.1016/j.tree.2019.03.006

51. Alhussein M, Aurangzeb K, Haider SI, 2020, Hybrid CNN-LSTM model for short-term individual household load forecasting. *IEEE Access*, 8:180544–180557.

https://doi.org/10.1109/ACCESS.2020.3028281

52. Tian C, Ma J, Zhang C, *et al*, 2018, A deep neural network model for short-term load forecast based on long short-term memory network and convolutional neural network[J]. *Energies*, 11(12):3493.

https://doi.org/10.3390/en11123493

53. Kim HY, Won CH, 2018, Forecasting the volatility of stock price index: A hybrid model integrating LSTM with multiple GARCH-type models. *Expert Syst Appl*, 103:25–37.

https://doi.org/10.1016/j.eswa.2018.03.002

54. Park K, Choi Y, Choi WJ, *et al.*, 2020, LSTM-based battery remaining useful life prediction with multi-channel charging profiles. *IEEE Access*, 8:20786–20798.

https://doi.org/10.1109/ACCESS.2020.2968939

55. Yu Y, Si X, Hu C, *et al.*, 2019, A review of recurrent neural networks: LSTM cells and network architectures. *Neural Comput*, 31(7):1235–1270.

https://doi.org/10.1162/neco_a_01199

56. Chimmula VKR, Zhang L, 2020, Time series forecasting of COVID-19 transmission in Canada using LSTM networks. *Chaos Solitons Fract*, 135:109864.

https://doi.org/10.1016/j.chaos.2020.109864

57. Shahid F, Zameer A, Muneeb M, 2020, Predictions for COVID-19 with deep learning models of LSTM, GRU and Bi-LSTM. *Chaos Solitons Fract*, 140:110212.

https://doi.org/10.1016/j.chaos.2020.110212

58. Zhao J, Mao X, Chen L, 2019, Speech emotion recognition using deep 1D & 2D CNN LSTM networks. *Biomed Signal Process*, 47:312–323.

https://doi.org/10.1016/j.bspc.2018.08.035

59. Kim T-Y, Cho S-B, 2019, Predicting residential energy consumption using CNN-LSTM neural networks. *Energy*, 182:72–81.

https://doi.org/10.1016/j.energy.2019.05.230

60. Graves A, Schmidhuber J, 2005, Framewise phoneme classification with bidirectional LSTM and other neural network architectures. *Neural Netw*, 18(5):602–610.

https://doi.org/10.1016/j.neunet.2005.06.042

61. Wang G, Hao J, Ma J, *et al.*, 2011, A comparative assessment of ensemble learning for credit scoring. *Expert Syst Appl*, 38(1):223–230.

https://doi.org/10.1016/j.eswa.2010.06.048

62. Xu R, Lin H, Lu K, *et al*, 2021, A forest fire detection system based on ensemble learning[J]. *Forests*, 12(2):217.

https://doi.org/10.3390/f12020217

63. Krawczyk B, Minku LL, Gama J, *et al.*, 2017, Ensemble learning for data stream analysis: A survey. *Inform Fusion*, 37:132–156.

https://doi.org/10.1016/j.inffus.2017.02.004

64. Dong X, Yu Z, Cao W, *et al.*, 2020, A survey on ensemble learning. *Front Comput*, 14(2):241–258.

https://doi.org/10.1007/s11704-019-8208-z

65. Mienye ID, Sun Y, 2022, A survey of ensemble learning: Concepts, algorithms, applications, and prospects. *IEEE Access*, 10:99129–99149.

https://doi.org/10.1109/ACCESS.2022.3207287

66. Mao S, Chen J-W, Jiao L, *et al.*, 2019, Maximizing diversity by transformed ensemble learning. *Appl Soft Comput*, 82:105580.

https://doi.org/10.1016/j.asoc.2019.105580

67. Wang G, Sun J, Ma J, *et al.*, 2014, Sentiment classification: The contribution of ensemble learning. *Decis Support Syst*, 57:77–93.

https://doi.org/10.1016/j.dss.2013.08.002

68. Hamori S, Kawai M, Kume T, *et al.*, 2018, Ensemble learning or deep learning? Application to default risk analysis. *J Risk Financ Manage*, 11(1):12.

69. Xu R, Lin H, Lu K, *et al.*, 2021, A forest fire detection system based on ensemble learning. *Forests*, 12(2):217.

70. Yang Y, Lv H, Chen N, 2022, A survey on ensemble learning under the era of deep learning. *Artif Intell Rev*: 1–45.

https://doi.org/10.1007/s10462-022-10283-5

71. Ghorbanzadeh O, Blaschke T, Gholamnia K, *et al,* 2019, Evaluation of different machine learning methods and

deep-learning convolutional neural networks for landslide detection[J]. *Remote Sens*, 11(2):196.

https://doi.org/10.3390/rs11020196

72. Khan A, Sohail A, Zahoora U, *et al.*, 2020, A survey of the recent architectures of deep convolutional neural networks. *Artif Intell Rev*, 53(8):5455–5516.

https://doi.org/10.1007/s10462-020-09825-6

73. Reina-Romo E, Mandal S, Amorim P, *et al.*, 2021, Towards the experimentally-informed in silico nozzle design optimization for extrusion-based bioprinting of shear-thinning hydrogels. *Front Bioeng Biotechnol*, 9:701778.

74. Allencherry J, Pradeep N, Shrivastava R, *et al.*, 2022, Investigation of hydrogel and gelatin bath formulations for extrusion-based 3D bioprinting using deep learning. *Procedia CIRP*, 110:360–365.

https://doi.org/10.1016/j.procir.2022.06.064

75. Xu H, Liu Q, Casillas J, *et al.*, 2022, Prediction of cell viability in dynamic optical projection stereolithography-based bioprinting using machine learning. *J Intell Manuf*, 33(4):995–1005.

https://doi.org/10.1007/s10845-020-01708-5

76. Wu D, Xu C, 2018, Predictive modeling of droplet formation processes in inkjet-based bioprinting. *J Manuf Sci E-T Asme*, 140(10):101007.

https://doi.org/10.1115/1.4040619

77. Lee J, Oh SJ, An SH, *et al.*, 2020, Machine learning-based design strategy for 3D printable bioink: Elastic modulus and yield stress determine printability. *Biofabrication*, 12(3):035018.

https://doi.org/10.1088/1758-5090/ab8707

78. Yamanishi C, Parigoris E, Takayama S, 2020, Kinetic analysis of label-free microscale collagen gel contraction using machine learning-aided image analysis. *Front Bioeng Biotechnol*, 8:582602.

79. Tourlomousis F, Jia C, Karydis T, *et al.*, 2019, Machine learning metrology of cell confinement in melt electrowritten three-dimensional biomaterial substrates. *Microsyst Nanoeng*, 5(1):1–19.

80. Safir F, Vu N, Tadesse LF, *et al.*, 2022, Detecting bacteria in multi-cellular samples with combined acoustic bioprinting and Raman spectroscopy. *arXiv preprint arXiv:2206.09304*.

81. Nadernezhad A, Groll J, 2022, Machine learning reveals a general understanding of printability in formulations based on rheology additives. *Adv Sci*, 9(29):2202638.

https://doi.org/10.1002/advs.202202638

82. Guan J, You S, Xiang Y, *et al.*, 2021, Compensating the cell-induced light scattering effect in light-based bioprinting using deep learning. *Biofabrication*, 14(1):015011.

https://doi.org/10.1088/1758-5090/ac3b92

83. Bone JM, Childs CM, Menon A, *et al.*, 2020, Hierarchical machine learning for high-fidelity 3D printed biopolymers. *ACS Biomater Sci Eng*, 6(12):7021–7031.

https://doi.org/10.1021/acsbiomaterials.0c00755

84. Shi J, Song J, Song B, *et al.*, 2019, Multi-objective optimization design through machine learning for drop-on-demand bioprinting. *Engineering*, 5(3):586–593.

https://doi.org/10.1016/j.eng.2018.12.009

85. Shi J, Wu B, Song B, *et al.*, 2018, Learning-based cell injection control for precise drop-on-demand cell printing. *Ann Biomed Eng*, 46(9):1267–1279.

https://doi.org/10.1007/s10439-018-2054-2

86. Fu Z, Angeline V, Sun W, 2021, Evaluation of printing parameters on 3D extrusion printing of pluronic hydrogels and machine learning guided parameter recommendation. *Int J Bioprint*, 7(4):434.

87. Tian S, Stevens R, McInnes BT, *et al.*, 2021, Machine assisted experimentation of extrusion-based bioprinting systems. *Micromachines*, 12(7):780.

88. Ruberu K, Senadeera M, Rana S, *et al.*, 2021, Coupling machine learning with 3D bioprinting to fast track optimisation of extrusion printing. *Appl Mater Today*, 22:100914.

https://doi.org/10.1016/j.apmt.2020.100914

89. Jin Z, Zhang Z, Shao X, *et al.*, 2021, Monitoring anomalies in 3D bioprinting with deep neural networks. *ACS Biomater Sci Eng*.

https://doi.org/10.1021/acsbiomaterials.0c01761

90. Bonatti AF, Vozzi G, Kai Chua C, *et al.*, 2022, A deep learning approach for error detection and quantification in extrusion-based bioprinting. *Mater Today*, 70:131–135.

https://doi.org/10.1016/j.matpr.2022.09.006

91. Tebon PJ, Wang B, Markowitz AL, *et al.*, 2022, Drug screening at single-organoid resolution via bioprinting and interferometry. *bioRxiv*, 2021:2021–2031.

92. Tröndle K, Miotto G, Rizzo L, *et al.*, 2022, Deep learning-assisted nephrotoxicity testing with bioprinted renal spheroids. *Int J Bioprint*, 8(2):528.