Int. J. Advance Soft Compu. Appl, Vol. 12, No. 2, July 2020
ISSN 2074-8523; Copyright © ICSRS Publication, 2020
www.i-csrs.org

Deep CNN Model for Nanotoxicity Classification Using Microscopic Images

Bhavna Saini, Sumit Srivastava and A.K.Bajpai

Manipal University Jaipur (Rajasthan), India
e-mail: bhavna.saini@jaipur.manipal.edu
Manipal University Jaipur (Rajasthan), India
e-mail: sumit.srivastava@jaipur.manipal.edu
Government Autonomous Science College, Jabalpur (MP) India

Abstract

Nanocarriers' usage turned out to be a transforming factor in the field of medical diagnosis and therapies. The drug delivery system is one of these therapies, where nanocarriers are used for controlled and targeted drug delivery to the diseased sites. During this process, the critical issue of toxicity can arise, which needs to be addressed well before. Computational modeling would be a beneficial asset for nanomaterials researchers because it can foresee the toxicity, rest on previous experimental data. This paper focuses on identifying the toxic nature of nanocarriers by computing the microscopic images and determining the dead cells count present on it. The current research work faces the problem of inadequacy of proper datasets, hence solving this problem by generating new images using Generative adversarial network (GAN), which further utilized to build a Convolutional neural network (CNN) based toxicity classifier.

Keywords: *Computational Modelling, Nanotoxicity, Drug Delivery System, Microscopic Images, GAN, CNN*

1 Introduction

Nanotechnology has prominently advanced in biomedical applications such as gene/drug delivery systems, probing of DNA structure, tissue engineering, etc. with the utilization of nanomaterials (NM). One of the prime factors that makes them accessible for use in therapeutic applications is their small size and surface features [1], [2]. The size and surface can be consumed in a controlled manner to make them eligible for designing synthetic NM using different chemical compositions. All these headways have brought about a very fast increase in the requisite of NMs in

biological applications. One of these popular applications is drug delivery process, which makes use of nanomaterials as nanocarriers.

Drug Delivery Process

Drug delivery is a therapeutic process that utilizes nanomaterials to transfer pharmaceutical compounds or drug inside the body and in a suitable amount to have the physiological effect [3]. The primary objective of the drug delivery process is to successfully incorporate the drug into nanocarriers and deliver it to the diseased site [3], [6], [8], [9]. Chemotherapy is one of the therapeutic processes, which follows the same procedure for the treatment of cancer or tumor [4], [5]. Inclusive of various compelling properties and features, the use of nanomaterials also gives rise to different harmful effects not only for humans but also for the environment [6]. The small size of NM exposes their large surface area to cellular units that can lead to some toxic effects. Another crucial situation that generally arises during the stabilization of NM for the drug delivery process is the use of reducing agents that may also contribute significantly to overall toxicity. Thus, for designing a drug delivery system, the toxicity is one of the critical issues to be addressed seriously, as the clinical trials are not much successful due to undetected toxic effects of employed nanomaterials [8].

Toxicity During Drug Delivery Process

The chemical compounds and drugs for the therapeutic process cannot be utilized directly due to their solubility and toxicity issues. If an excess dose of drugs is used, it can lead to severe physiological discomforts. To address this issue, a wide variety of biopolymers, superparamagnetic NPs are used as carriers in drug delivery process [7]. These biopolymers are completely non-toxic, readily available, and economically feasible, but they are soluble in water. Hence to make them water-insoluble, crosslinking is done. Among various preparation methods of biopolymer nanoparticles, emulsion crosslinking is one of the efficient techniques that make use of certain crosslinking agents for biopolymers to make them insoluble in water [10], [11]. In fact, biopolymers are not toxic in nature, but the use of chemical crosslinking agents may cause toxicity in the prepared nanoparticle. The crosslinking agents combine with biopolymers and make them insoluble, which can also result in toxicity generation. This toxic nature needs to address well before its actual utilization.

Computational Modelling

Computational assessment of biological images has pulled in great interest of analysts worldwide with the introduction of machine learning techniques [12], [13]. For an effective prediction, it is highly required to identify the significant features or patterns, which correctly characterize the biological images used. Thus being

motivated by the prediction potential of cytotoxicity pertaining to nanocarriers, the objectives of the present work include identifying the toxic nature of some gelatin-based nanocarriers that have been previously investigated as drug delivery nanocarriers. The toxicity prediction was performed on microscopic images collected from Govt. Autonomous Science College, Jabalpur. The dataset contained 15 images, which were generated during the drug delivery process. Performing accurate toxicity prediction and classification on small and imbalanced image dataset is quite challenging. To solve this problem, this paper proposed a semi-supervised learning along with GAN (Generative Adversarial Networks).

2 Related Work

Automated detection of cells is nowadays a quite studied field, for which an enormous number of methods have been defined [14] - [17]. In 2016, Atteya et al. [18] have aimed to describe an advanced vision-based system to automatically detect, classify, and track the organic cells using a recently developed SOPAT-System (Smart On-line Particle Analysis Technology), a photo-optical image acquisition device combined with innovative image analysis software. The proposed method includes image de-noising, binarization and Enhancement, as well as object recognition, localization and classification based on the analysis of particles' size and texture. Watershed transformation is used to extract manually connected objects, and it produced best accuracy. However, it can easily fall into the local optima.

In 2020, Huang et al. [19] have proposed a novel blood cell classification framework based on medical hyperspectral imaging, which combined a Modulated Gabor Wavelet (MGW) and deep Convolutional neural network (CNN) kernels, named as MGCNN. For each Convolutional layer, multi-scale and orientation Gabor operators were taken dot product with initial CNN kernels. The essence was to transform the Convolutional kernels into the frequency domain to learn features. MGCNN is having the better classification performance than traditional CNNs and widely used support vector machine approaches. Yet, it is computationally expensive.

Moreover, SVM [20], [21] is utilized to handle the classification problem of leukocyte, it attains good accuracy when compared over other conventional algorithms, it has high recognition and classification accuracy, and it categorizes the texture features into distinct classes of hardwood species. However, there are few conflicts with SVM such as performance will not be good for huge datasets, and for noisy data, also it will not perform well. Random Forest [22] was used to analyze the texture of mineral surfaces, and it has good classification accuracy. However, it is more complex to implement.

The proposed work focuses on the area, where cell detection and estimation plays an essential role in toxicity prediction of drug delivery nanocarriers.

3 Problem Formulations or Methodology Materials

The gelatin used as a biopolymer for the preparation of nanocarriers was supplied by Loba Chemie, India. Glutaraldehyde was used as a crosslinking agent for gelatin and purchased from Loba Chemie, India, and used without further purification. Other chemicals such as toluene (for preparing oil phase) and acetone were of analytical purity grade.

Methods

Preparation of Gelatin Nanocarriers:

The gelatin nanocarriers were prepared by the emulsion crosslinking method, as described elsewhere [10]. In a typical experiment, 2 g of gelatin was dissolved in 20 mL hot water under mild stirring for one hr till the complete dissolution of gelatin. Now to this aqueous solution of gelatin, a definite volume of toluene was added with vigorous stirring so that a stable emulsion of gelatin was obtained. Now a 1:1 (v/v) mixture of toluene and glutaraldehyde was prepared with constant stirring and added to the gelatin emulsion with a pre-determined rate. The crosslinking of gelatin with glutaraldehyde was allowed to take place for 6 h till the color of gelatin starts changing from slightly yellow to dark brown, which was an indication of crosslinking of gelatin by glutaraldehyde. After completion of the crosslinking reaction, the gelatin nanoparticles were filtered, washed thoroughly with acetone and water, and dried. The dried gelatin nanoparticles were stored in air-tight containers for further study.

In Vitro Toxicity of Gelatin Nanoparticles

In in-vitro methods, toxicity is judged based on the fastening of cells. If a chemical compound is capable of modifying the morphology of cells, then it unfavorably influences the cell growth and causes the cell death. To examine in-vitro cytotoxicity of the as-prepared glutaraldehyde cross-linked gelatin nanoparticle, the extract method (ISO – 1099 33-5) on L 929 fibroblast cells has been widely used [23]. In this method, the powdered material is immersed in culture medium with serum, and then the extract is build up by incubating the pre penetrated test material with serum for 24 h. After the incubation process, the extract is filtered and diluted with culture medium to get the required concentration. Later on, after dilution of test sample extract, positive control, and negative control are investigated microscopically for the cellular reaction. The microscopic images indicate toxicity behavior. The toxicity levels, such as mild, moderate, and severe, are determined using the counts of live and dead cells shown in the microscopic images. The manual count of these cells can lead to error. Therefore, the tool of image processing can be successfully and precisely used for accurately calculating the drug potency and toxicity of chemical compounds, which act as a promising drug candidate.

4 The Proposed Method

The proposed work has been divided into three phases, which are as follows:

1. Initially, a set of image processing techniques were applied to process the images and to identify the number of dead cells present in the images. Based on dead cells count, the images were labeled as toxic or non-toxic.

2. GAN was configured to generate artificial labeled data having similar distribution as that of original data. These newly created images would considerably improve classification accuracy.

3. Further DCNN (Deep Convolutional Neural Network) was implemented to classify the unlabeled images as toxic or non-toxic, based on the original images and GAN generated images together.

4.1 Identification of Dead Cells using Image Processing Techniques

The automatic cell detection system is a quite studied field, where several methods had been defined, and still, the study of new techniques is going on [24] – [27]. The work done in the past mostly focuses on cell detection using various methods like FCNN, FCRN, UCNN [28] – [30], but nowhere toxicity is related to cell detection. This paper focuses on a relatively new or undiscovered area, where toxicity prediction of nanocarriers is made based on cell detection and evaluation.

For this, first, the Vitro cytotoxicity test is performed on L-929 fibroblast cells. The reactivity of the test is evaluated using microscopic images generated through three different control samples, which are: positive control, negative control, and test sample. The grade value associated with each sample helps to determine the toxic nature. The highest value, i.e., 4 tends to toxic nature while the lowest one, i.e., 0 is non-toxic. These grade values are decided based on dead cells generated in each sample during the cytotoxicity test. This paper focuses on positive control on positive control images, as shown in Figure 1. The next section discusses the algorithm applied.

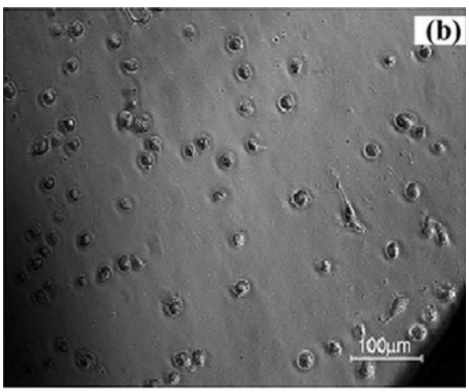

Fig 1: Positive Control Image

4.1.1 Positive Control images (Class PoC)

Initially, the discontinuity in intensity value of images is identified, which depends upon the direction of maximum intensity variation, base image position, and the change in contrast to the local image along normal. The gradient magnitude was identified at each pixel by a convolving image with vertical and horizontal derivative filters as mentioned in Equation 1-3.

Gradient Vector

$$\nabla I = \left[\frac{\partial I}{\partial x}, \frac{\partial I}{\partial y}\right]^T, \tag{1}$$

*w*here T = Threshold Value using sobel kernel,
$\partial \mathrm{x} = f(x+a, y) - f(x-a, \mathrm{y})$ and $\partial \mathrm{y} = f(x, y+a) - f(x, y+a)$,
*w*here a = small integer value.

Magnitude

$$|\nabla I| = \sqrt{\left(\frac{\partial I}{\partial x}\right)^2 + \left(\frac{\partial I}{\partial y}\right)^2} \tag{2}$$

Orientation

$$\theta = \alpha \tan\left(\frac{\partial \mathrm{I}}{\partial \mathrm{x}}, \frac{\partial \mathrm{I}}{\partial \mathrm{y}}\right), \text{where } \alpha = \text{arc.} \tag{3}$$

Fig 2(a) shows an example of results in accurate gradient results.

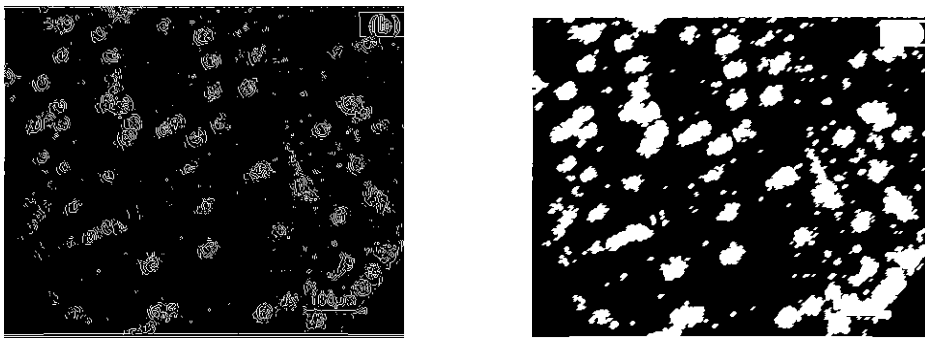

Figure 2: (a) Result after applying gradient vector on the image and (b) the resultant image after hole filling process.

Generating Structuring Element

A structuring element S_L is required to dilate the images; hence, a structuring element was generated evenly to adjacent centers. The dilation process widen out the foreground objects which help in easy cell detection as shown in Equation 4-5.

$$(I \oplus S_L)(x,y) = max\{A(x-x', y-y') + S_L(x',y') | (x',y') \in D\} \tag{4}$$

As we are using a flat structuring element, hence

$$S_L(x',y') = 0 \text{ and} \tag{5}$$
$$(I \oplus S_L)(x,y) = max\{A(x-x', y-y') + (x',y') \in D\}$$

To make the cell appearance more clear [31], hole filling is done using Equation 6

$$F(x,y) = \begin{cases} 1 - I(x,y), & \text{If } (x,y) \text{ is on the border of } I \\ x, & \text{Otherwise} \end{cases} \tag{6}$$

After dilation process, hole filling is done which makes the cells more visible as shown in Fig 2(b). The resultant image is enhanced now, but to remove noise and make it clearer filtering is done.

Top-hat Filtering

The filtering method includes a morphological opening that is performed using the structuring element. It extracts the small details or components from the given image using Equation 7. It is implemented as a difference between input image I and the structuring element.

$$I_{w(f)} = f - S_L \tag{7}$$

Fig 3 shows the outcome of the above function.

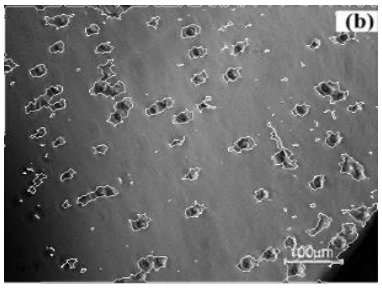

Figure 3: Results after morphological operation

Most of the cells in the image were identified, but still, the count of dead cells was incorrect. The variance in the count was due to closely connected cells, which were considered as one object, as shown in Fig 4. To separate these connected cells, segmentation methods are used.

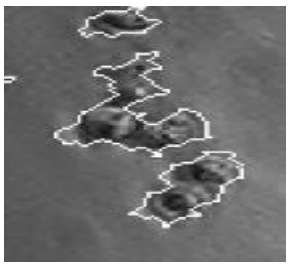

Figure 4: Clustered Cells

Watershed Algorithm is one of the classical segmentation method used in the past [32], [33]. Various improvements and variations of this method were generated and utilized due to its simplicity, speed, and easily adjustable distance map factor [34], [35], [36].

Watershed Algorithm

Initially, a global thresholding was applied, which was further followed by the watershed algorithm to separate the overlapped cells and to determine the correct cell count. To find out the optimum threshold, which isolated the background and foreground pixels, Otsu's method was used [37]. It is a nonparametric and unsupervised method that analyses for the threshold that mitigates the intraclass variance. The weighted within-class variance is calculated as mentioned below in Equation 8.

$$\sigma_w^2 = \mathrm{p}(t)\sigma_1^2(t) + p_2(t)\sigma_2^2(t) \quad (8)$$

Where p is the class probability of different grey level pixels

And value of p is given by applying Equation 9:

$$p_1(t) = \sum_{i=0}^{t} P(i)$$
$$p_2(t) = \sum_{i=0}^{255} P(i) \quad (9)$$

Total variance is calculated as mentioned in Equation 10

$$\sigma_2 = \sigma_w^2(t) + \sigma_b^2(t) \quad (10)$$

Where $\sigma_w^2(t)$ = within class variance

$\sigma_b^2(t)$ = between class variance

Distance transform method of watershed was used for cell segmentation. Initially the distance has been calculated from each pixel to its nearest neighbor pixel having non zero values.

Let P be the set of points in space Z. For every point p of P, the distance d (p) of p to the corresponding set Pc is defined as in Equation 11

$$\forall p \in P, d(p) = dist(p, P^c) \quad (11)$$

Where $dist(p, P^c)$ is the distance of p to the nearest point P^c

A section of distance d at level is defined in Equation 12

$$Y_i(d) = \{p: d(p) \geq i\} = P \ominus D_i \quad (12)$$

Where D_i is a disk of radius i

The resultant image after distance based watershed segmentation algorithm is shown in Fig 5.

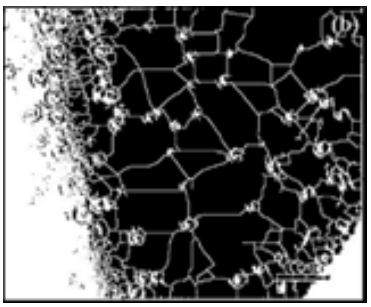

Figure 5: Image after segmentation

The distance segmentation method resulted in suitable partition of the connected cells. Fig 6 illustrated the outcome with an example.

Figure 6: Cells after segmentation

The algorithm was tested on total 15 positive control images and it works with approx. 97% accuracy. Table 1 illustrates the result of all images.

Table 1: Positive Control Images Class Result

Image ID	Total Cells	Correctly Identified	Undetected Cells	Intruded Cells	Precision	Recall
1	68	67	1	0	1.00	0.98
2	64	64	0	0	1	1
3	67	65	1	1	0.98	1
4	47	47	0	0	1	1
5	38	37	1	0	0.97	1
6	35	34	1	0	1	0.97
7	37	35	0	2	0.95	1
8	46	43	3	0	0.93	1
9	64	59	3	2	0.96	0.95
10	68	67	1	0	1	0.98
11	64	62	2	0	1	0.96
12	60	58	0	2	0.96	1
13	71	69	0	2	0.97	1
14	45	45	0	0	1	1
15	65	62	1	2	0.96	0.98

4.2 Generation of New Images using GAN

The author named Goodfellow et al. [38], proposed Generative Adversarial Network (GAN), which is based on the unsupervised model. It follows a sequential approach, where at every step, a set of techniques are applied. Every step feeds forwards the other until the resultant images occur. GAN consists of a generator neural network and an adversarial neural network. A generator is a de-convolutional neural network, which expands the images instead of shrinking. The filters are move over input images, and new images having higher spatial dimensions are

generated. The generator automatically discovers the uniformity or patterns present in input images and learns by itself. This learning further helps to generate new images from original images. These newly generated images are as alike as the original one, which makes it difficult to distinguish between the two [39], [40].
In the past GAN, techniques had been used in various biomedical researches for improving the classification of images [41], [42]. Yang J. et al. [43] implemented GAN for data augmentation of X-ray prohibited item images. Other than, these various authors had used this technique to generate new images [44], [45].
The discriminator learns the features of original images and finds out what all features have a major role in making it real. Finding out the same, it gives feedback to the generator based on which it generates new similar images. The discriminator is a standard convolution network, which distinguish between real and fake images. Here the focus is only on the generator part, as the goal is to generate more images to increase the dataset size. Figure 7 shows the basic steps followed in GAN to generate more microscopic images.

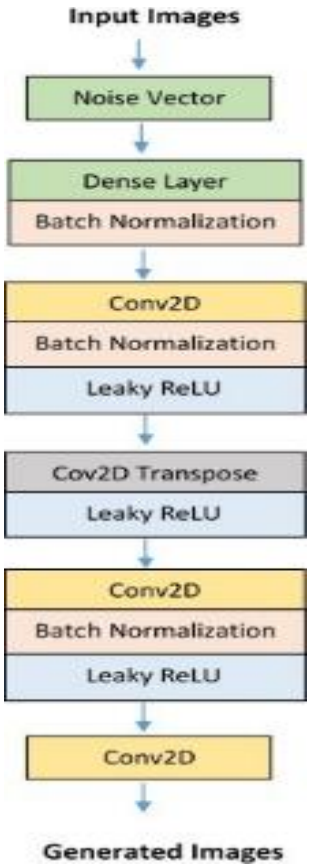

Figure 7: Step of GAN to generate new images

The section below explains each step.

Noise Vector

Initially, a noise vector has been added to the input images. With the addition of a random noise vector, the class of resultant image becomes different from that of original images. A small noise addition to the image does not affect its appearance much but modifies its configuration. An example of the same is shown in the figure 8 below.

Adding such a small value vector does not make any changes in the appearance of the original image.

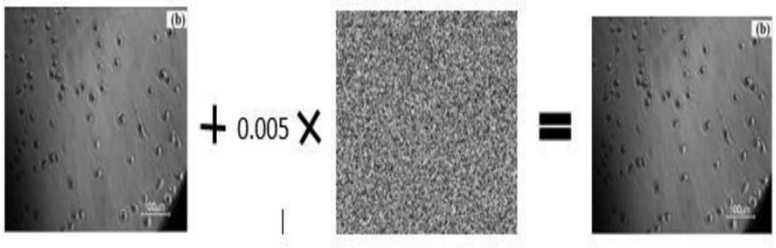

Figure 8: Addition of noise vector to input image

Dense Layer

The generator network works as a black box, which takes normally generated number vectors as input and generate new images. To generate such a required architecture, a dense layer is required. This dense layer creates a dense vector which is further converted to image as shown in figure 9.

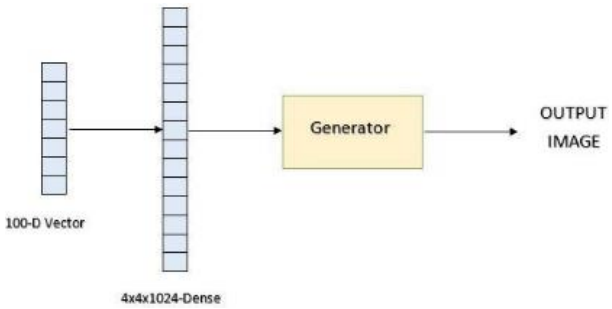

Figure 9: Dense Layer

Batch Normalization

Batch Normalization is one of the key phases of GAN. It allows each network layer to learn by itself. Here, each layer learns a bit more independently of other layers.

It checks for normalization of activation function and makes sure it does not go too high or low. This allows the network to train on new features, which could not get trained earlier. Batch normalization also solves the problem of overfitting.

Convolution Layer

As explained in the above section, it is a kind of mathematical function, where combining two variables generates a new third variable. Here, the first input refers to the original image, while the second one is a small filter or kernel. The kernel slides along the original image and works on the small blocks at a time. This sliding process obtains a new image.

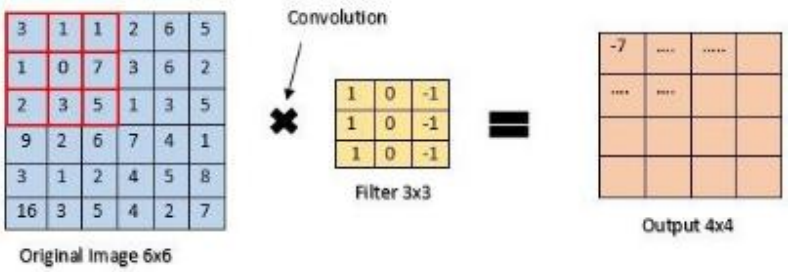

Figure 10: Convolutional Layer

LeakyReLU

LeakyRelU refers to Leaky version of a Rectified Linear Unit and is one of the most popular activation functions. The ReLU function can be defined as max(0,Z). It means the range of this function lies between 0 and Z. Here, every activation value which occurs in region < 0, becomes zero automatically. This problem is known as dying ReLU and can be overcome by applying LeakyReLU. LeakyReLU solves the problem by using a constant and small non-zero gradient value.

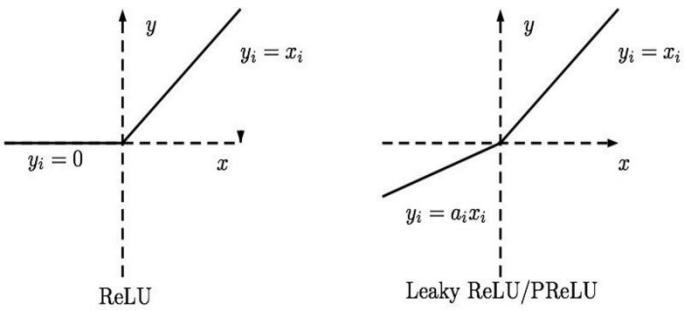

Figure 11: ReLU and LeakyReLU functions

The above mentioned multi-layer step processing of GAN is applied in the work to increase the dataset size of microscopic images. The next section discusses the algorithm used for training as well as generating new images.

Results after GAN Implementation

Development of nanotoxicity prediction models is becoming increasingly important in the risk assessment of engineered nanomaterials. However, it has significant obstacles caused by the wide heterogeneities of published literature in terms of data completeness and quality. Over the last decade, substantial advances have been made in various computer vision technologies and many of them are based on Convolutional Neural Network (CNN) architecture. Typically, slow convergence and the need for extensive parameter tuning adversely affect a stochastic gradient descent algorithm using Back-Propagation (BP) trains CNN, but the training process. This work is focused on the nanotoxicity prediction using the microscopic images. As the manually collected images are in small count, it will be improved by Generative Adversarial Networks (GAN), which increases the dataset size based on the configuration of original images.
GAN is implemented to increase the dataset size by generating more microscopic images. Initially, 35 microscopic images were taken as input. As mentioned in algorithm 1, these input images IL are defined as labeled images. Here labeled images refer to the images which are already classified as toxic based on dead cells count. The generator phase of GAN automatically learns the feature of input images and generates 150 more images. These newly generated images are named as fabricated images IF in the algorithm.

Algorithm 1: Image Generation using GAN

Input: $\mathbf{I_{labelled}} \cup \mathbf{I_{Unlabelled}}$, **Toxlearner1** and **Toxlearner2**, **GAN**
(Where $\mathbf{I_{labelled}}$ = dataset images labelled as Toxic, $\mathbf{I_{Unlabelled}}$ = unlabeled dataset images)

Output: Toxic Image Classifier TC

Initialize: $\mathbf{I_L} = \mathbf{I_{labelled}}$, $\mathbf{I_U} = \mathbf{I_{Unlabelled}}$, $\mathbf{I_F} = \emptyset$ **(**where $\mathbf{I_F}$ = fabricated labeled dataset**)**

1. while $\mathbf{I_F}$ is transformed
2. $\mathbf{I_G}$ = **GAN(** $\mathbf{I_L}$) (where $\mathbf{I_G}$ = GAN generated images using original labeled images)
3. Train **Toxlearner1** & **Toxlearner2** using $\mathbf{I_L} + \mathbf{I_G}$ as dataset

4. Apply Toxlearner1 and predict labels for $\mathbf{I_U}$ **as** $\mathbf{I_{label-1}}$
5. Apply Toxlearner1 and predict labels for $\mathbf{I_U}$ **as** $\mathbf{I_{label-2}}$
6. for each sample of $\mathbf{I_U}$ check
7. **if** $\mathbf{I_{label-1}(i) = = I_{label-2}(i)}$
8. Add unlabeled instance **i** to fabricated class
9. Add $\mathbf{I_U(i)} = I_F$
10. end if
11. end for
12. $\mathbf{I_L = I_L + I_F}$ (Final labeled dataset having fabricated data included)
13. $\mathbf{I_U = I_U - I_F}$
14. end while
15. $\mathbf{I_G = GAN(I_L)}$, $\mathbf{I_L = I_L + I_G}$
16. Apply Classifier **TC** on $\mathbf{I_L}$

During training, a GAN framework has two loss functions one is for generator, and another one is for the discriminator. The authors have applied 1500 epochs in the model. The accuracy and loss functions vary with the increase in the number of epochs. As can be seen in the Figures 12-14, on 300 epochs, the pixels are scattered, and the authors could not figure anything out from it. After 900 epochs, the pixels seem more organized and the loss functions for both discriminator and generator are getting decreased. With the increase in epochs i.e. on 1500 epochs the images get more clear and are quite similar to that of original images. The variation in the loss has decreased gradually and becomes almost constant at the end of training. There exist a minor change in the loss value of both generator and discriminator, which indicates equilibrium. Here, a total of 150 images are generated, which are used in developing a CNN model for toxicity classification**.**

```
Epoch 0100 [D loss: 0.098, acc.: 098.2%] [G loss: 3.399]
Epoch 0200 [D loss: 0.272, acc.: 089.1%] [G loss: 2.390]
Epoch 0300 [D loss: 0.308, acc.: 087.7%] [G loss: 2.795]
```

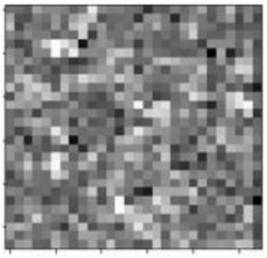

Figure 12: Results after 300 epochs

```
Epoch 0700 [D loss: 0.308, acc.: 087.7%] [G loss: 2.795]
Epoch 0800 [D loss: 0.372, acc.: 084.0%] [G loss: 2.897]
Epoch 0900 [D loss: 0.303, acc.: 088.2%] [G loss: 2.476]
```

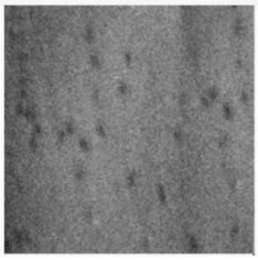

Figure 13: Results after 900 epoch

```
Epoch 1300 [D loss: 0.489, acc.: 075.6%] [G loss: 1.415]
Epoch 1400 [D loss: 0.485, acc.: 076.1%] [G loss: 1.412]
Epoch 1500 [D loss: 0.478, acc.: 077.0%] [G loss: 1.437]
```

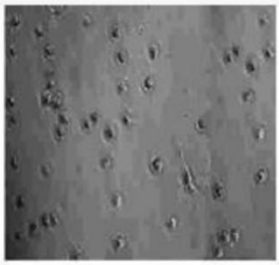

Figure 14: Results after 1500 epoch

4.3 Classification using CNN Model

The dataset having both the original and fabricated images is transferred to the CNN model for classification. The dataset is split into 80% training data and rest 20% for validation to have a generalized model. The layers of the CNN models are shown in Table 2.

Input Layer:
In the input layer, the image size is 28*28*1, which corresponds to height, width, and channel size. Here, value 1 represents the gray-scale image.

Convolution Layer:
The CNN model has a total of 3 convolution layers. The first layer applied 32 filters with filter size as 3*3, the second layer has 64 filters, and the third layer has 128 filters. All these filters are of size 3*3. Along with these convolution layers, three max-pooling layers are also added with filter size as 2*2.

Table 2: CNN Layers

Layer (type)	Output Shape	Parameters
Conv2d_1 (Conv2D)	(None, 28, 28, 32)	320
LeakyRelu_1	(None, 28, 28, 32)	0
Max_pooling2d_1	(None, 14, 14, 32)	0
Conv2d_2 (Conv2D)	(None, 14, 14, 64)	18496
LeakyRelu_2	(None, 14, 14, 64)	0
Max_pooling2d_2	(None, 7, 7, 64)	0
Conv2d_3 (Conv2D)	(None, 7, 7, 128)	73856
LeakyRelu_3	(None, 7, 7, 128)	0
Max_pooling2d_3	(None, 4, 4, 128)	0
Flatten_1	(None, 2048)	0
Dense_1	(None, 128)	262272
LeakyRelu_4	(None, 128)	0
Dense_2	(None, 2)	258
Total params: 355,202		
Trainable params: 355,202		
Non-trainable params: 0		

In the last dense layer, classification is done that works well and achieves an accuracy of 92% and validation loss as 0.43.

The acceptance of deep learning methods in the area of nanoscience could be increased due to its success in other areas. But the major challenge faced by deep CNN models in the biomedical field is the limited amount of data. The limited amount of training data cannot generalize the features accurately and results in overfitting. Another issue associated with the training of deep CNN models from scratch is its high computational power, memory, and time. In this case, transfer learning and fine-tuning can prove helpful. Hence, in future work, transfer learning can be used for nanotoxicity classification.

Conclusion

The present study has developed an image analysis framework which works on the microscopic images generated during drug delivery process, to identify the toxic nature of the nanocarriers used. There are numerous approaches defined in the past for automatic cell detection, but to the best of authors knowledge there is no work exist for toxicity prediction during drug delivery process based on cell detection. This work presents an unsupervised approach to enhance the dataset size. In drug

delivery process there is a need to evaluate the toxic nature of nanocarriers. This toxicity can be identified by counting the dead cells present in microscopic images which are generated during drug delivery testing. For this, an image processing framework was designed. To bring in more reliability and accuracy in the model, Convolutional Neural Network (CNN) had been implemented preceded by Generative adversarial networks. Generative Adversarial Networks (GAN) was found to solve the problem of small dataset by generating additional images based on the configuration of original images. All these images collectively formed a dataset to further classify the test images using CNN.

REFERENCES

1. Alagarasi, A. (2011). Introduction to nanomaterials. National Center for Environmental Research, 141-198.
2. Abhilash, M. (2010). Potential applications of Nanoparticles. International Journal of Pharma and Bio Sciences, 1(1), 1-12.
3. Lehner, R., Wang, X., Marsch, S., &Hunziker, P. (2013). Intelligent nanomaterials for medicine: carrier platforms and targeting strategies in the context of clinical application. Nanomedicine: Nanotechnology, Biology and Medicine, 9(6), 742-757.
4. De Jong, W. H., & Borm, P. J. (2008). Drug delivery and nanoparticles: applications and hazards. International journal of nanomedicine, 3(2), 133.
5. Tiwari, G., Tiwari, R., Sriwastawa, B., Bhati, L., Pandey, S., Pandey, P., & Bannerjee, S. K. (2012). Drug delivery systems: An updated review. International journal of pharmaceutical investigation, 2(1), 2.
6. Agarwal, M., Murugan, M. S., Sharma, A., Rai, R., Kamboj, A., Sharma, H., & Roy, S. K. (2013). Nanoparticles and its Toxic Effects: A Review. Int. J. Curr. Microbiol. App. Sci, 2(10), 76-82.
7. Chomoucka, J., Drbohlavova, J., Huska, D., Adam, V., Kizek, R., & Hubalek, J. (2010). Magnetic nanoparticles and targeted drug delivering. Pharmacological research, 62(2), 144-149.
8. Fischer, H. C., & Chan, W. C. (2007). Nanotoxicity: the growing need for in vivo study. Current opinion in biotechnology, 18(6), 565-571.
9. Tiwari, G., Tiwari, R., Sriwastawa, B., Bhati, L., Pandey, S., Pandey, P., &Bannerjee, S. K. (2012). Drug delivery systems: An updated review. International journal of pharmaceutical investigation, 2(1), 2.
10. Pal, A., Bajpai, J., & Bajpai, A. K.(2018) Easy fabrication and characterization of gelatin nanocarriers and in vitro investigation of swelling controlled release dynamics of paclitaxel. Polymer Bulletin, 1-21.
11. Khan, H., Shukla, R. N., & Bajpai, A. K. (2016). Genipin-modified gelatin nanocarriers as swelling controlled drug delivery system for in vitro release of cytarabine. Materials Science and Engineering: C, 61, 457-465.

12. Sharma, M., Sharma, S., & Singh, G. (2018). Performance Analysis of Statistical and Supervised Learning Techniques in Stock Data Mining. Data, 3(4), 54.
13. Shen, D., Wu, G., & Suk, H. I. (2017). Deep learning in medical image analysis. Annual review of biomedical engineering, 19, 221-248.
14. Xing, F., & Yang, L. (2016). Robust nucleus/cell detection and segmentation in digital pathology and microscopy images: a comprehensive review. IEEE reviews in biomedical engineering, 9, 234-263.
15. Di Ruberto, C., Dempster, A., Khan, S., & Jarra, B. (2002). Analysis of infected blood cell images using morphological operators. Image and vision computing, 20(2), 133-146.
16. Ambriz-Colin, F., Torres-Cisneros, M., Avina-Cervantes, J. G., Saavedra-Martinez, J. E., Debeir, O., & Sanchez-Mondragon, J. J. (2006, November). Detection of biological cells in phase-contrast microscopy images. In 2006 Fifth Mexican International Conference on Artificial Intelligence (pp. 68-77). IEEE.
17. Byun, J., Verardo, M. R., Sumengen, B., Lewis, G. P., Manjunath, B. S., & Fisher, S. K. (2006). Automated tool for the detection of cell nuclei in digital microscopic images: application to retinal images. Mol Vis, 12(105-07), 949-60.
18. Atteya, M. A., Salem, M. A. M. M., Hegazy, D. A. K. M., & Roushdy, M. I. (2016). Image segmentation and particles classification using texture analysis method. Research on Biomedical Engineering, 32(3), 243-252.
19. Huang, Q., Li, W., Zhang, B., Li, Q., Tao, R., & Lovell, N. H. (2019). Blood Cell Classification Based on Hyperspectral Imaging With Modulated Gabor and CNN. IEEE journal of biomedical and health informatics.
20. Duan, Y., Wang, J., Hu, M., Zhou, M., Li, Q., Sun, L., ... & Wang, Y. (2019). Leukocyte classification based on spatial and spectral features of microscopic hyperspectral images. Optics & Laser Technology, 112, 530-538.
21. Yadav, A. R., Anand, R. S., Dewal, M. L., & Gupta, S. (2015). Gaussian image pyramid based texture features for classification of microscopic images of hardwood species. Optik, 126(24), 5570-5578.
22. Camalan, M., Çavur, M., & Hoşten, Ç. (2017). Assessment of chromite liberation spectrum on microscopic images by means of a supervised image classification. Powder Technology, 322, 214-225.
23. ISO 10993-5: Biological evaluation of medical devices—Part 5: Tests for in vitro cytotoxicity. Arlington, VA: ANSI/AAMI; 1999.
24. Xing, F., & Yang, L. (2016). Robust nucleus/cell detection and segmentation in digital pathology and microscopy images: a comprehensive review. IEEE reviews in biomedical engineering, 9, 234-263.
25. Di Ruberto, C., Dempster, A., Khan, S., &Jarra, B. (2002). Analysis of infected blood cell images using morphological operators. Image and vision computing, 20(2), 133-146.

26. Ambriz-Colin, F., Torres-Cisneros, M., Avina-Cervantes, J. G., Saavedra-Martinez, J. E., Debeir, O., & Sanchez-Mondragon, J. J. (2006, November). Detection of biological cells in phase-contrast microscopy images. In Artificial Intelligence, 2006. MICAI'06. Fifth Mexican International Conference on (pp. 68-77). IEEE.
27. Byun, J., Verardo, M. R., Sumengen, B., Lewis, G. P., Manjunath, B. S., & Fisher, S. K. (2006). Automated tool for the detection of cell nuclei in digital microscopic images: application to retinal images. Mol Vis, 12(105-07), 949-60.
28. Tan, R. X., Zhang, J., Chen, P., Wang, B., & Xia, Y. (2018, August). Cells Counting with Convolutional Neural Network. In International Conference on Intelligent Computing (pp. 102-111). Springer, Cham.
29. Xie, W., Noble, J. A., & Zisserman, A. (2018). Microscopy cell counting and detection with fully convolutional regression networks. Computer methods in biomechanics and biomedical engineering: Imaging & Visualization, 6(3), 283-292.
30. Xue, Y., Ray, N., Hugh, J., & Bigras, G. (2016, October). Cell counting by regression using convolutional neural network. In European Conference on Computer Vision (pp. 274-290). Springer, Cham.
31. Somasundaram, K., &Kalaiselvi, T. (2010). A method for filling holes in objects of medical images using region labeling and run length encoding schemes. In National Conference on Image Processing (NCIMP) (pp. 110-115).
32. Lin, G., Adiga, U., Olson, K., Guzowski, J. F., Barnes, C. A., &Roysam, B. (2003). A hybrid 3D watershed algorithm incorporating gradient cues and object models for automatic segmentation of nuclei in confocal image stacks. Cytometry Part A, 56(1), 23-36.
33. Mackin, J. R., Newton, L. M., Turner, J. N., &Roysam, B. (1998). Advances in high-speed, three-dimensional imaging and automated segmentation algorithms for thick and overlapped clusters in cytologic preparations. Application to cervical smears. Analytical and quantitative cytology and histology, 20(2), 105-121.
34. Ortiz De Solórzano, C., Garcia Rodriguez, E., Jones, A., Pinkel, D., Gray, J. W., Sudar, D., & Lockett, S. J. (1999). Segmentation of confocal microscope images of cell nuclei in thick tissue sections. Journal of Microscopy, 193(3), 212-226
35. Al-Kofahi, Y., Lassoued, W., Lee, W., &Roysam, B. (2010). Improved automatic detection and segmentation of cell nuclei in histopathology images. IEEE Transactions on Biomedical Engineering, 57(4), 841-852.
36. Wang, Y., Zhang, Z., Wang, H., & Bi, S. (2015). Segmentation of the clustered cells with optimized boundary detection in negative phase contrast images. PloS one, 10(6).
37. Otsu, N. (1979). A threshold selection method from gray-level histograms. IEEE transactions on systems, man, and cybernetics, 9(1), 62-66.

38. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., ... & Bengio, Y. (2014). Generative adversarial nets. In Advances in neural information processing systems (pp. 2672-2680).
39. van Tulder, G., & de Bruijne, M. (2016). Combining generative and discriminative representation learning for lung CT analysis with convolutional restricted boltzmann machines. IEEE transactions on medical imaging, 35(5), 1262-1272.
40. Salakhutdinov, R. (2015). Learning deep generative models. Annual Review of Statistics and Its Application, 2, 361-385.
41. Onishi, Y., Teramoto, A., Tsujimoto, M., Tsukamoto, T., Saito, K., Toyama, H. & Fujita, H. (2019). Automated Pulmonary Nodule Classification in Computed Tomography Images Using a Deep Convolutional Neural Network Trained by Generative Adversarial Networks. BioMed research international, 2019.
42. Dong, X., Lei, Y., Wang, T., Thomas, M., Tang, L., Curran, W. J., ... & Yang, X. (2019). Automatic multiorgan segmentation in thorax CT images using U-net-GAN. Medical physics, 46(5), 2157-2168.
43. Yang, J., Zhao, Z., Zhang, H., & Shi, Y. (2019). Data augmentation for X-ray prohibited item images using generative adversarial networks. IEEE Access, 7, 28894-28902.
44. Dunn, I., Melham, T., & Kroening, D. (2019). Generating Realistic Unrestricted Adversarial Inputs using Dual-Objective GAN Training. arXiv preprint arXiv:1905.02463.
45. Vandenhende, S., De Brabandere, B., Neven, D., & Van Gool, L. (2019, May). A three-player GAN: generating hard samples to improve classification networks. In 2019 16th International Conference on Machine Vision Applications (MVA) (pp. 1-6). IEEE.

Notes on contributors

Dr. Bhavna Saini is Assistant Professor at the Department of Information Technology, Manipal University Jaipur. She has completed her masters from Central University of Rajasthan with specialization in Information Security in 2013. Her area of interest are data analytics and its applications, cloud computing and security.

Dr. Sumit Srivastava Professor in Department of Information Technology, Manipal University Jaipur. He has published more the 50 research papers in peer-reviewed International Journals, Book series and IEEE/Springer/ Elsevier conference. He is an expertise in the domain of Data Analytics and Image Processing. He is also a senior member of IEEE with more than 14 years of teaching experience. He has organized many

national and international conference under IEEE & Springer banner in Rajasthan. He also the publication chair of two consecutive editions of IEEE conference on MOOC, Innovation and Computing (MITE). Beside this he also Coordinated & mentored the Bilateral Exchange between the HEIG-VD university and Manipal University Jaipur for Two consecutive years.

Prof.A.K.Bajpai had done his Ph.D. (Polymer chemistry) and D.Sc. from Rani Durgawati University, Jabalpur. He is having 30 years of undergraduate and postgraduate teaching and research experience. The areas on which he is working includes Synthesis of biomedical polymers, drug delivery through nanocarriers, nanocomposites, stimuli responsive drug delivery systems. He is a regular reviewer of several journals like Elsevier, Wiley, Taylor and Francis, ACS and other international publications and had guided 62 candidates for Ph.D. degree. One patent on water remediation has been filed in collaboration with the DRDO, New Delhi. Major research projects by leading national funding agencies like DRDO, Department of Science and Technology, University Grants Commission, Board of Research in Nuclear Sciences, Department of Atomic Energy. He is also an author of book “Responsive Drug Delivery: From Theory to Applications”, By Smihter Rapra Publications (UK).