An Artificial Intelligence powered Digital Inline Holographic Microscopy and characterization scheme.

Rajkumar Vaghashiya1*, Varun Chauhan1*, Kaushal Kapadiya1*, Smit Sanghavi1*, Ishita Nandwani1, Jeanie Jessica Basumatary1, Riya Thakore1, Dongmin Seo2, Sungkyu Seo3, and Mohendra Roy4**

1Dept. of Computer Science and Engineering, Pandit Deendayal Petroleum University (PDPU), Gandhinagar, India 382007
2Korea Research Institute of Ships & Ocean Engineering, Daejeon, South Korea 34103
3Dept. of Electronics and Information Engineering, Korea University, South Korea 30019
4Dept. of Information and Communication Technology, Pandit Deendayal Petroleum University (PDPU), Gandhinagar, India 382007
**Email: mohendra.roy@sot.pdpu.ac.in

Abstract:

Digital Inline Holography (DIH) based microscopy is a well-established technique for the characterization of nano and microparticles, such as biological cells, artificial microparticles, quantum dots, etc. Due to its simplicity and cost-effectiveness, various practical solutions such as auto characterization of complete blood count (CBC), cell viability test, and 3D cell tomography have been developed. In our previous work, we demonstrated the feasibility of this system to perform complete blood count along with the auto characterization of cell-lines as well as shape and size characterization of the microparticles. However, its performance suffered due to the weak signals from some of the cells owing to their poor signatures and the presence of background noise. The auto characterization technique therein was based on the parameters determined from our empirical findings, which limit the system in terms of its cell-line recognition power. In this work, we try to address these issues by leveraging an artificial intelligence-powered auto signal enhancing scheme as well as adaptive cell characterization technique. The performance comparison of our proposed method with the existing analytical model shows an increase in accuracy to >98% along with the signal enhancement of >5 dB for most cell types like Red Blood Cell (RBC) and White Blood Cell (WBC), except the cancer cells (HepG2 and MCF-7) for which the accuracy is about 84%.

* Contributed equally
Introduction:

Digital Inline Holography (DIH) is a proven technology that has been used for the development of various imaging modalities, especially for the cell and microparticle imaging\(^1\). The technique is widely popular for its simple arrangement. It comprises a lens-less detector such as a charged coupled device (CCD) or complementary metal-oxide semiconductor (CMOS) sensor, and a semi-coherent light source, such as light emitting diode (LED) or LED laser. The simplicity of the arrangement is as depicted in figure 1(b). Since this arrangement consists of a few components, most of which are easily available and at a low price, therefore it reduces the overall cost of the system\(^2\). The absence of lens or other optical arrangements except for the detector, the light source, and the sample holder allows it to fit into a very small space, thereby reducing the size of the overall system. This facilitates the feasibility of the system for the applications such point of care system\(^3\). Together with the auto characterization modality, this system presents a perfect platform for the development of cost-effective telemedicine systems\(^4,5\).

Recent advancements in machine learning, especially deep learning, have facilitated many applications concerning medical diagnostics\(^6-12\), and have been widely adopted in the field of microscopy\(^13-15\). Recently, deep learning has been incorporated with the digital inline holography\(^14\), where it is has been used to enhance the resolution of the DIH micrographs\(^16\) and enabled polarization-based holographic microscopy\(^17\).

In our previous work, we have successfully developed the DIH imaging system for the complete blood count (hemocytometry) using an analytical model based on handcrafted features\(^3\) that can automatically segment out the individual cells from a whole frame DIH micrograph and subsequently analyze them based on the handcrafted parameters. However, the performance of the system is dependent on the uniform illumination as well as signatures of the microparticle samples. Though the influences of these in the overall performance of the system are very less, these issues need to be addressed to get a better performance. Further, the handcrafted approach of finding the features for every additional cell line is time-consuming and prone to subjective errors. To address these limitations, in this work we have developed an artificial intelligence (AI) powered signal enhancement scheme for the DIH micrographs that can enhance the signal quality (signal to noise ratio (SNR)) for various cell lines in a heterogeneous cell sample. For this we used the autoencoder based denoising scheme\(^18\).
Further we have developed the auto characterization method based on convolutional neural network\textsuperscript{19,20} (CNN) architecture to classify the various cell lines from the DIH micrograph.

In this article, we have described the detailed methods adopted for the designing as well as optimization of various parameters to devise a suitable model with better accuracy. These optimized models are simple, lightweight, and require a smaller number of samples for effectively learning the cell signatures. The details are as given in the following sections.

Figure 1. Schematic of the DIH microscope setup and the proposed neural network architecture for the auto characterization of DIH micrographs. (a) schematic of the principle of diffraction signature generation of a micro object, (b) schematic of the DIH imaging setup which shows the simplicity of the setup, (c) schematic of the dataset creation process by cropping individual cell diffraction pattern from the whole frame DIH micrograph, (d) the schematic of the proposed denoising and classification architecture. Here, the denoising autoencoder enhances the signal of the individual cells which is then fed to the CNN module for classification.
**Methods:**

**DIH imaging setup:**
The schematic of our proposed setup is as shown in figure 1(b). When light from the coherent or semi-coherent source passes through the micro object, it produces characteristic diffraction pattern of the object\(^1\),\(^2\),\(^2\) as shown in figure 1(a). These diffraction patterns are prominent just beneath the sample, typically a few millimeters away from the sample plane, from where they are captured using a high-density image sensor such as CCD or CMOS\(^2\). As these signatures are significant enough to be captured by the bared image sensor, it does not require any kind of lens arrangement\(^3\). In our proposed setup, we used a semi-coherent LED light source with a peak wavelength of 470 ± 5 nm (HT-P318FCHU-ZZZZ, Harvatek). The diffraction patterns were captured using a 5-megapixel CMOS image sensor (EO-5012M, Edmund Optics), and C-Chip (C10228) from Invitrogen was used to hold the cell samples\(^2\),\(^5\). All these components can fit in a compact dimension of 12 cm × 9.0 cm × 9.0 cm.\(^2\),\(^4\) Due to the absence of a lens-based setup, the field of view of this system is about 20 times to that of a conventional optical microscope at 100X. This high throughput provides an extra advantage to characterize several cells within one single frame.

**Preparation of various cell lines:**
In this work we used various cell lines, starting from red blood cell (RBC), white blood cell (WBC), the cancer cell lines HepG2 (human liver cell-line) and MCF7 (human breast cancer cell-line), and artificial glass beads of 10 μm and 20 μm. The preparations of these cell lines are as follows\(^2\),\(^3\).

**RBC:** The RBC samples were prepared from the whole blood samples that were collected from the Korea University Hospital under IRB approval. The samples were diluted about 16,000 times by using RPMI solution (Thermo Scientific)\(^2\),\(^3\).

**WBC:** First, Ficoll solution (Ficoll-Paque\(^{™}\) Plus, GE Healthcare, Chicago, IL, USA) was used to isolate mononuclear cells (PBMCs) from the whole blood. The samples of PBMCs obtained using the Ficoll solution are mixtures of lymphocytes and monocytes. To separate these two cell types, the MACS (Magnetic-activated cell sorting) method was applied. The helper-T cells in the lymphocytes were separated using the CD4 antibody (#130-090-877, Miltenyi Biotec, Germany), and the cytotoxic-T cells with the CD8 antibody (# 130-090-878, Miltenyi Biotec,
Germany), Miltenyi Biotec, Germany). Finally, 10 µL of this solution was then loaded into the unruled C-Chip cell counting chamber.

**HepG2:** The HepG2 cell lines were prepared from the American Type Culture Collection (ATCC HB-8065) and incubated in a high-glucose medium (Dulbecco Modified Eagle Medium, DMEM) with 10% heat-inactivated fetal bovine serum, 0.1% gentamycin, and a 1 penicillin/streptomycin solution under 95% relative humidity and 5% CO\textsubscript{2} at 37\degree C. The developed cells were then trypsinized and separated from 24 well plate and incubated from 2-5 min at 37\degree C. These cells were then diluted with DMEM solution\textsuperscript{2,3}.

**MCF7:** The MCF7 cell samples were prepared from the American Type Culture Collection (ATCC HTB-22). The cells were preserved in a solution of DMEM containing 1% penicillin/streptomycin solution, 0.1% gentamycin, and 10% calf serum at 95% relative humidity and 5% CO\textsubscript{2} at 37\degree C. These cells were then trypsinized and separated from the 24 well plate. These separated cells were then incubated for 2-5min at 37\degree C. The cells were then washed with DMEM solution. 10 µL of this solution was then loaded in the C-Chip\textsuperscript{2,3}.

**10 and 20 micron bead:** The 10 µm bead and 20 µm bead samples were prepared by diluting the respective polystyrene microbeads (Thermo Scientific) with de-ionized water\textsuperscript{2,3}.

The use of human whole blood in the experiment was approved by the Institutional Review Board (Approval No. MD17020-001) of Korea University Guro Hospital (Guro-gu, Seoul, Korea).

**Dataset creation:** A whole frame DIH image contains an average of 1000 diffraction patterns of microparticles. Deep learning-based architectures utilize the features of each class, and typically require a minimum of a few hundred diffraction patterns of each cell type for optimal learning. Therefore, we cropped individual diffraction pattern of each cell type with a window of 66 × 66 pixels as shown in figure 1(c). We further augment this base sample set by rotating the individual diffraction patterns with an increasing angle of 10 degrees clockwise. Finally, a dataset of 1980 samples for each of the cell lines and microparticles was created, totalling to 11880 samples for all the 6 classes under study. Similarly, we created the dataset for 60 × 60, 56 × 56, 50 × 50, 46 × 46, 40 × 40, and 36 × 36 cell sizes as input sets, with each set further divided into training and test folds. As the data augmentation used sample rotation, the splitting of the dataset into the train, validation and test folds needs to be done while keeping a check on data leakage. Augmented samples distributed across train and test set may bias the
model and may give a wrong estimate of its performance as the test data may not of entirely “unseen” samples. Accounting for this, the 1980 samples of each class were carefully split into 1490 training samples, 166 validation samples, and 324 testing samples.

**Denoising modality:**

For denoising of the DIH micrographs, we adopted the concept of autoencoder\(^\text{24}\). An autoencoder is an unsupervised scheme that scuffles to recreate the input at its output. It consists of an input layer \((x)\), an output layer \((r)\), and a hidden layer \((h)\). The hidden layer \(h\) termed as a code layer stands for the input in a reduced dimension. The whole network structure can be labelled into two parts. The first part is an encoder, which tries to code the input as \(h = f(x)\), and the second part is a decoder which tries to recreate the input from the reduced code layer as \(r = g(h)\), where \(r\) is the recreated assortment of input \(x\) (See figure 1(d)). Basically, it tries to attain \(r = g(f(x))\). However, this is not a linear transformation since model is enforced to learn the significant features of the input to encode it into the code layer (in reduced dimension).

In this work, we specifically used the denoising version of the autoencoder. Traditionally, the autoencoders try to reduce the loss as \(L(x, g(f(x)))\). However, the denoising autoencoder attempts to reduce the cost as \(L(x, g(f(x')))\). Where \(x'\) is the noisy form of the input \(x\).

In this work we tried two different methods to design the denoising architectures, namely, extreme learning machine (ELM) and convolutional neural network (CNN).

**ELM:** This is a single hidden layer fully connected architecture\(^\text{25}\). In this method, the input weights are initiated randomly and kept intact. Only the output weights take part in the learning process through a straightforward learning contrivance\(^\text{25–27}\). For \(N\) arbitrary input samples \(x_i \in \mathbb{R}^n\) and their counterpart targets \(t_i \in \mathbb{R}^m\), the ELM achieves this mapping using the following relation as shown in equation (1).

\[
H\beta = T
\]  
(1)

Here, \(H\) is the hidden layer output matrix, \(\beta\) is the output weight matrix, i.e. between the hidden layer and the output layer, and \(T\) is the target matrix or matrix of desired output\(^\text{26}\). From equation (1), we can obtain the \(\beta\) using Moore–Penrose pseudoinverse\(^\text{25}\) as shown in equation (2).

\[
\beta = (H^T H)^{-1} H^T T
\]  
(2)

In the extended sequential learning form of ELM, the \(\beta\) can update sequentially. This provides an added advantage of updating the learning whenever a new type of sample is available, thus
providing the flexibility of transfer of learning. The $\beta$ update mechanism\textsuperscript{28,29} is as shown in equation (3).

$$\beta_n = \beta_{n-1} + P_n^{-1}(T_n - H_n\beta_{n-1})H_n^T$$  \hspace{1cm} (3)

Here

$$P_n = P_{n-1} + H_nH_n^T$$  \hspace{1cm} (4)

For $n = 1$,

$$P_{n-1} = P_0 = \left(\frac{1}{c} + H_0H_0^T\right)$$  \hspace{1cm} (5)

Here $H_0$ is the hidden layer output with the 1\textsuperscript{st} sample or 1\textsuperscript{st} batch of samples\textsuperscript{30}.

\textbf{CNN:} Convolutional neural networks (CNN)\textsuperscript{6,19,20} are a type of neural network widely used in the analysis of spatial data such as image classification and object segmentation. In this network, two dimensional kernels are used to extract the spatial features from the input patterns, using a convolution operation between the kernel and the input. The typical architecture of a CNN is as shown in figure 1(d). Here, the kernel is shared spatially by the input or by the feature map. The feature at the location $(i, j)$ in the $k^{th}$ feature map of the $l^{th}$ layer can be evaluated as shown in equation (6).

$$Z_{i,j,k}^l = (W_{k}^l)^T X_{i,j}^l + b_k^l$$  \hspace{1cm} (6)

Here, $W_k^l$ and $b_k^l$ are the weights and the bias vector of the $k^{th}$ filter in the $l^{th}$ layer. Here the weight layer is shared spatially which reduces the complexity. $X_{i,j}^l$ is the value of the input at location $(i, j)$ of the $l^{th}$ layer. The nonlinearity in this network can be obtained by introducing the activation function, denoted here as $g(.)$. The activated output can be represent as shown in equation (7).

$$a_{i,j,k}^l = g(Z_{i,j,k}^l)$$  \hspace{1cm} (7)

Additionally, there are pooling layers that introduce shift-invariance by reducing the resolution of the activated feature maps. Each pooling layer connects the feature map to the preceding convolutional layer. The expression for pooling is as shown in equation (8).

$$y_{i,j,k}^l = p(a_{n,m,k}^l), \forall (m,n) \in R_{ij}$$  \hspace{1cm} (8)

Here $p(.)$ is a pooling operation for local neighbourhood $R_{ij}$ around the location $(i, j)$. 
In our work, we used CNN for both denoising as well as classification. The details of their architectures and their impacts are discussed in the result section. The detailed description of the CNN is given in the supplementary section.

Results and Discussion:

Performance of denoising algorithms:

For efficient and adaptive denoising, we analysed various autoencoder schemes, starting with the fully connected autoencoder. In our first iteration, we experimented with the fully connected network having three hidden layers with 512, 256, and 512 number of neurons respectively. The input layer is the 1D vectorized array of the input image (e.g. of $66 \times 66$ pixels). The input to the model is the noisy version of the input image and the expected target output is the original image. The noisy images were created using a Gaussian distribution with variance ranging from 100 to 600 with zero mean (refer the supplementary section for detail).

Further, we experimented with an increased network size having five hidden layers with 256, 128, 64, 128, and 256 number of neurons respectively. In all these networks, rectified linear unit (ReLU) was used as the activation function while mean squared error (MSE) was used to calculate the loss. The Adam optimizer (after iterations over other optimizers) was found to deliver better convergence and hence used to perfect the weight and biases. The denoising performance was quantified in terms of the improvement in SNR, measured in dB, denoted here by $SNR_{imp}$, as given by equation (9).

$$SNR_{imp} = SNR_{out} - SNR_{in}$$

where $SNR_{out} = 10 \log_{10} \left( \frac{\sum_{i=1}^{N} x_i^2}{\sum_{n=1}^{N} (\tilde{x}_i - x_i)^2} \right)$, and $SNR_{in} = 10 \log_{10} \left( \frac{\sum_{n=1}^{N} x_i^2}{\sum_{n=1}^{N} (\tilde{x}_i - x_i)^2} \right)$. Here $x_i$ is the value of sampling point $i$ in the original DIH signal, $\tilde{x}_i$ is the value of sampling point $i$ in the noisy DIH, and $\tilde{x}_i$ is the value of sampling point $i$ in the denoised version of the same image. $N$ is total number of sample points in that DIH image.

The fully connected network for both the above configuration shows no significant improvement in $SNR_{imp}$ after reaching saturation at around -10.08 dB. For further improvement, we experimented with CNN architecture using various models with a different number of convolution layers and distinct kernel sizes. The configuration of the model which accomplished the best outcomes is $3 \times 3, 3 \times 3, 5 \times 5, 5 \times 5, 7 \times 7, 7 \times 7, 1 \times 1$ with 32 filters in each layer except the last layer. The last layer consists of a single pixel filter ($1 \times 1$ filter) that
is used to condense the output across all the 32 filters. Here, the input and output size are the same. Padding was used to maintain the original size after the output of each convolutional layer. Adam optimizer was used to optimize the network to reduce the mean squared error loss. The CNN results show a better reconstruction as shown in figure 2.

The CNN network has been optimized for various parameters. First, the optimization of the network for various design parameters, such as varying the convolution layers and the kernel sizes, was carried out. The results in figure 3(a) show that the architecture with kernel sizes $3 \times 3$, $3 \times 3$, $5 \times 5$, $5 \times 5$, $7 \times 7$, $7 \times 7$, $1 \times 1$ has a better performance in terms of SNR$_{imp}$. The performance of the optimized network for various noise parameters, as shown in figure 3(b), indicates the network performs better reconstruction with increasing noise variance in the image. An increase in the variance results in a noisier image, which warrants a detailed reconstruction to reverse it to the original form, and hence larger the value of SNR$_{imp}$. Therefore, a higher improvement in SNR$_{imp}$ implies the network has learnt the optimal representational features for the cell types which enables it to perform a better qualitative reconstruction. Figure 3(c) compares the reconstruction performance of the model on different sizes of the input image. Due to the black-box nature of deep learning methods, we had to create datasets with multiple

![Figure 2. Reconstructed results from the optimized CNN. The top row is the original DIH image of a single RBC, WBC, MCF7, HepG2, 10µm bead, and 20µm bead. The middle row is the noisy version (with variance 100) of the corresponding original images. The third row is the reconstructed version of the corresponding original images from the noisy image. The fourth row is the 2D intensity contour plot of the original image to show the unique signature of each of these cell lines.](image-url)
cell-signature dimensions, such that the smallest size just covered the central signature of the cell and increasing the window size till it covered significant background portion as well. The models were evaluated across varying cell sizes to determine the optimal signal to background ratio, the spatial extent up to which the models covered the features, and to study its effects on the model performance. This analysis is critical in understanding the model explainability and interpretability since having a size larger than the optimum increases the inclusion of background artifacts that affect denoising as well as overpower the cell signal while having a smaller one could exclude the important deterministic features of the cell signature. The convergence in the training phase of the network is as shown in figure 3(d). The results depict the loss across the first epoch, with a high variation in the initial phase, gets smoother towards the end of the first iteration. The advantage of this system is that it generalizes well for all the types of cell lines using the same model.
Figure 3. Results from the CNN and ELM autoencoder. (a) The performance of the CNN autoencoder for improved SNR (average of all the classes) across various layers and kernel sizes. (b) The performance of the CNN autoencoder across varying noise levels. Here, the variance of the Gaussian noise ranges from 100 to 600, and evaluated on the optimal network architecture i.e. 3,3,5,5,7,7,1. (c) The performance of the CNN autoencoder across various input sizes (cropping size). Here, the sizes vary from 66×66 to 36×36. (d) The convergence of the optimal CNN network with the number of samples for the first epoch. (e) The number of hidden layer neurons in ELM autoencoder vs improvement in SNR. (f) Variance vs improved SNR for ELM autoencoder. (g) Input size vs improved SNR for ELM autoencoder, and (h) The convergence of ELM autoencoder within the 1st epoch.
Further, we tried the ELM architecture which is well known for its fast convergence\textsuperscript{25}. The results in figure 3(e-h) show the performance of the ELM architecture with varying number of neurons in the hidden layer. As it can be concluded from figure 3(e), the model with 2000 neurons provides better performance in terms of SNR\textsubscript{imp}. Further, the optimized model has been used to test the performance across various noise levels as shown in figure 3(f). It is observed that the model maintains the SNR\textsubscript{imp} value on increasing the noise in the input image i.e. the image quality of the output relative to the input remains the same. The results of the model performance across different sizes of the input image, as shown in figure 3(g), indicate that the 40 × 40 is having higher value of SNR. However, the variation is of 2 as compared to the variation for the size 50 × 50, that is of 1.5, which is the lowest compared to all the other sizes. Since CNN shows a substantial performance with lower variance for the 50 × 50 input size, therefore, we fixed it as optimal for all the further study and comparisons. Figure 3(h) shows the loss across the first epoch for ELM which is remarkably high initially but converges faster, as compared to CNN, after training with only a few thousand samples. This faster convergence may help save time and resources during incremental training phases for newer cell types. The performances of these optimized models have been compared with the traditional denoising methods (as shown in Table 1). It is concluded from the data that CNN shows better performance compared to the other modalities (The details of the traditional methods are provided in the supplementary document). Therefore, we prefer to use CNN for the denoising.

\textbf{Table 1:} Comparison of improved SNR with respect to the variance for various denoising modalities (here we keep input and output size as 50×50).

| Variance | Gaussian | Average | Median | Bilateral | BM3D | CNN | ELM |
|----------|----------|---------|--------|-----------|------|-----|-----|
| 100      | 4.52558  | 4.237827| 3.199475| -0.02635  | 5.684597| 6.06905| 5.79161|
| 200      | 4.613044 | 4.198564| 3.068866| -0.0064   | 5.713797| 7.38544| 6.31896|
| 300      | 4.476552 | 4.259405| 3.078091| -0.10943  | 5.119646| 7.26874| 6.57256|
| 400      | 4.674922 | 4.703331| 3.40547 | -0.05448  | 3.792226| 7.71265| 6.36135|
| 500      | 4.128021 | 4.201562| 3.198964| -0.14542  | 2.206876| 7.72364| 6.09254|
| 600      | 4.023621 | 4.183404| 2.908946| -0.16924  | 1.68329 | 7.978772| 6.19359|

\textbf{Performance of classification algorithm:}

Since the diffraction patterns of cells and microparticles in a DIH micrograph depend upon their physical and optical properties, therefore, the diffraction patterns carry the unique signatures of each of the cell types as shown in the 2D contour plot in figure 2. These unique signatures can be utilized for the classification of these cell types. Since our previous inference
concludes that CNN works better for denoising, therefore we experimented with the same modality for the classification as well. In this work, in order to determine the optimal architecture of CNN for cell-line recognition, we first proceeded to find the optimal depth of the network by studying the classification performance of the model on increasing the depth, by adding convolutional and pooling layers, as well as by varying number of kernels and kernel size, till we reached performance saturation. We have experimented and evaluated various shallow and deep CNN models to classify cell lines. The details of the model architecture are as described in figure 4.

Figure 4. The CNN Architectures with varying depth. The models in the solid box are for the optimization of the model with varying depth. The models inside the dotted box are for the optimization of the parameters. Here the orange line is the input layer and the rectangle represent the 2D convolution layer. The kernel size and number of kernels are indicated inside the round brackets, the dropout rate in square brackets, and the number of neurons in fully connected layers are placed directly within the rectangle. The aqua blue, green and purple lines are the max-pool layers. The red line is the output layer and uses a SoftMax activation function. The final optimized architecture is as represented by the last 3D figure.

The Deep Model starts with a convolutional (Conv2D) layer having 512 kernels of size $3 \times 3$, followed by max-pooling layer of same kernel size. The output from this is further convoluted with 128 kernels of $3 \times 3$ size with dropout rate of 0.5, and then max pool with $2 \times 2$ kernel. This output goes to Conv2D layer with 64 of $3 \times 3$ sized kernels and a dropout rate of 0.2. We further reduce the dimension using a $2 \times 2$ max pool kernel. The output of this layer further
convolves with 32 of $3 \times 3$ kernels, then a dropout of 0.2. The output dimension from this convoluted layer is further reduced by using the max pool with a $3 \times 3$ kernel. This again convolves with 16 of $3 \times 3$ kernels, and a max pool layer with $3 \times 3$ kernel. The output of which is again convoluted with 8 of $3 \times 3$ kernels, followed by a $3 \times 3$ max pool. This output is then vectorized and input to a fully connected (FC) layer having 256 nodes, and then to another FC with 128 nodes and having a dropout of 0.2. The final layer is a SoftMax function, with 6 output nodes. The model architectures used for studying the impact of the network depth on performance are well described in figure 4 (first model inside the solid box). Similarly, the other model for studying impact of network breadth is as depicted in figure 4 (inside the solid box). Once the approximate optimal depth and breadth had been determined, we proceeded to fine-tune the hyper-parameters such as the number of kernels, kernel size and dropouts, to reach the best performance of the models across varying cell sizes (input dimension of cells). The more stable architectures with optimized hyperparameters are as shown in figure 4 (inside the dotted box). In all the models, AdaDelta\textsuperscript{38} provided better convergence as compared to other optimizers and has been used as the model optimizer, with categorical cross-entropy\textsuperscript{39,40} as a loss estimator. The result for depth and breadth optimization is as shown in figure 5(a-c), indicate that the average accuracy of the intermediate model is $>0.85$ (including all the classes), whereas it is $<0.85$ for shallow and deep networks. Therefore, we proceed with this intermediate model as an optimum model for further study. Considering the intermediate model has the optimum breadth and depth, the further optimization of the parameters and the results are as shown in figure 5(d-g).
Figure 5. Results for the optimization of the CNN model. (a-c) shows the performance of shallow, intermediate, and deep networks respectively. (d-g) Performance of the fine-tuned intermediate architecture. (h) The confusion matrix, showing classification accuracy on the test dataset across all cell lines using the fine-tuned and optimized model. (i) The receiver operating characteristic (ROC) curve for each of the cell lines for the optimized model.

From the figure 5(d-g), it is inferred that Model3 shows better classification performance, on the validation fold, of all models. The results depict that there is consistency in performance for the input sizes 40 × 40 to 66 × 66, with a very small variance in the accuracy. The performance of this optimized model is further evaluated over the test dataset containing 324 samples of each cell type. The per-class performance of this model is shown in the confusion matrix of figure 5(h). The results depict that the model can classify RBC, WBC, 10 µm, and 20 µm bead with over 99% accuracy. However, the comparatively poor performance of about 90% for the cancer cells, HepG2 and MCF7, can be attributed to the non-homogeneity in their signature characteristics as well as the lack of sufficient original samples which further complicates the issue. This is well depicted in our previous work4 (see figure 2 of the reference).

From the receiver operating characteristic (ROC) curve for all the cell lines shown in the figure 5(i), the area under the curve (AUC) for all the cell lines is >0.99, except MCF7 (~0.95) and HepG2 (~0.96). From these results, it can be inferred that the classifier is working well, especially for RBC, WBC, 10 µm and 20 µm beads.
Additionally, we also investigate the transfer of learning to gauge the ability of the trained network to adapt to newer cell types. For this scenario, the CNN was initially trained with all the cell lines except RBC. From the per-class test accuracy shown in figure 6(c), it is observed that the model misclassified all the RBC samples as WBC. In the transfer of the learning phase, the initially trained network is frozen except for the last layer, which is modified to accommodate the newer class and kept trainable. The network is then re-trained with a mix of RBC samples. The per-class test accuracy of the re-trained model is shown in the confusion matrix of figure 6(d), where it is inferred that the re-trained network can classify RBC correctly with substantial accuracy.

From the epoch versus accuracy graph in figure 6(a), the transfer training achieved higher accuracy with the same number of epochs compared to the initial training. This is also validated by the epoch versus loss graph in figure 6(b). From these results, it can be inferred that the network can be effectively used to adapt to newer cell lines with very less amount of training.

Figure 6. The results from the transfer of learning study. (a) epoch vs. accuracy graph for the initial epochs in the initial training phase without RBC and then the transfer of training with RBC. The blue and saffron colour represent the initial training accuracy and validation curves. The green and red lines represent the transfer learning accuracy and validation curves, (b) epoch vs. loss, (c) the confusion matrix from the pre-trained model, (d) the confusion matrix with the re-trained model.
Conclusion:

In conclusion, we have explored the advantage of neural networks in the characterization of DIH micrographs. Here, we have perfected neural networks that can automatically improve the signal quality and classify the cell types. We find that this neural network can classify the RBC, WBC with great accuracy i.e. over 98% and the cancer cell with an accuracy of about 84%. This network is also flexible for adapting to newer cell lines by retraining the trained network with very few samples. Together with this algorithm, the lightweight and cost-effective DIH setup can be utilized as a point of care system for the diagnosis of pathological disorders in the resource-limited setup of our world.

Acknowledgement: This work has been carried out in collaboration with the Nano Bio Photonics lab of Korea University, South Korea and the Pandit Deendayal Petroleum University (PDPU), Gandhinagar, India. The part of this project at PDPU (the AI software development part) is supported by the PDPU seed grant of No. ORSP/R&D/PDPU/2019/MR/RO051.

Conflict of interest: The authors declare that they do not have any conflicting interests.

Reference:

1. Mudanyali, O. et al. Lensless On-chip Imaging of Cells Provides a New Tool for High-throughput Cell-Biology and Medical Diagnostics. e1650 (2009) doi:doi:10.3791/1650.

2. Roy, M. et al. Low-cost telemedicine device performing cell and particle size measurement based on lens-free shadow imaging technology. Biosens. Bioelectron. 67, 715–23 (2015).

3. Roy, M., Jin, G., Seo, D., Nam, M.-H. & Seo, S. A simple and low-cost device performing blood cell counting based on lens-free shadow imaging technique. Sensors Actuators B Chem. 201, 321–328 (2014).

4. Roy, M. et al. Automated Micro-Object Detection for Mobile Diagnostics Using Lens-Free Imaging Technology. Diagnostics 6, 17 (2016).

5. Roy, M. et al. Staining-free cell viability measurement technique using lens-free
shadow imaging platform. *Sensors Actuators, B Chem.* **224**, 577–583 (2015).

6. Liu, Y. *et al.* Detecting Cancer Metastases on Gigapixel Pathology Images. (2017).

7. Yu, K. H., Beam, A. L. & Kohane, I. S. Artificial intelligence in healthcare. *Nature Biomedical Engineering* **2**, 719–731 (2018).

8. Liu, Y. *et al.* Artificial intelligence–based breast cancer nodal metastasis detection insights into the black box for pathologists. *Arch. Pathol. Lab. Med.* **143**, 859–868 (2019).

9. Ardila, D. *et al.* End-to-end lung cancer screening with three-dimensional deep learning on low-dose chest computed tomography. *Nat. Med.* **25**, 954–961 (2019).

10. Im, H. *et al.* Design and clinical validation of a point-of-care device for the diagnosis of lymphoma via contrast-enhanced microholography and machine learning. *Nat. Biomed. Eng.* **2**, 666–674 (2018).

11. Wang, P. *et al.* Development and validation of a deep-learning algorithm for the detection of polyps during colonoscopy. *Nat. Biomed. Eng.* **2**, 741–748 (2018).

12. Ballard, Z. S. *et al.* Deep learning-enabled point-of-care sensing using multiplexed paper-based sensors. *npj Digit. Med.* **3**, 1–8 (2020).

13. Rivenson, Y. *et al.* Deep learning microscopy. *Optica* **4**, 1437 (2017).

14. Ozcan, A. Deep learning-enabled holography (Conference Presentation). in *Label-free Biomedical Imaging and Sensing (LBIS) 2020* (eds. Shaked, N. T. & Hayden, O.) vol. 11251 60 (SPIE, 2020).

15. Wang, H. *et al.* Deep learning enables high-throughput early detection and classification of bacterial colonies using time-lapse coherent imaging (Conference Presentation). in *Optics and Biophotonics in Low-Resource Settings VI* (eds. Levitz, D. & Ozcan, A.) vol. 11230 13 (SPIE, 2020).

16. de Haan, K. *et al.* Enhancing resolution in coherent microscopy using deep learning (Conference Presentation). in *Quantitative Phase Imaging VI* (eds. Popescu, G., Park, Y. & Liu, Y.) vol. 11249 3 (SPIE, 2020).

17. Liu, T. *et al.* Deep learning-based holographic polarization microscopy. (2020).

18. Kunapuli, S. S., Bh, P. C. & Singh, U. Enhanced Medical Image De-noising Using Auto Encoders and MLP. in *Communications in Computer and Information Science*
19. Indolia, S., Goswami, A. K., Mishra, S. P. & Asopa, P. Conceptual Understanding of Convolutional Neural Network- A Deep Learning Approach. in Procedia Computer Science vol. 132 679–688 (Elsevier B.V., 2018).

20. Yamashita, R., Nishio, M., Do, R. K. G. & Togashi, K. Convolutional neural networks: an overview and application in radiology. Insights into Imaging vol. 9 611–629 (2018).

21. Seo, S., Su, T.-W., Tseng, D. K., Erlinger, A. & Ozcan, A. Lensfree holographic imaging for on-chip cytometry and diagnostics. Lab Chip 9, 777–87 (2009).

22. Mudanyali, O. et al. Compact, light-weight and cost-effective microscope based on lensless incoherent holography for telemedicine applications. Lab Chip 10, 1417–1428 (2010).

23. Jin, G. et al. Lens-free shadow image based high-throughput continuous cell monitoring technique. Biosens. Bioelectron. 38, 126–31 (2012).

24. Ca, P. V., Edu, L. T., Lajoie, I., Ca, Y. B. & Ca, P.-A. M. Stacked Denoising Autoencoders: Learning Useful Representations in a Deep Network with a Local Denoising Criterion Pascal Vincent Hugo Larochelle Yoshua Bengio Pierre-Antoine Manzagol. Journal of Machine Learning Research vol. 11 (2010).

25. Cambria, E. et al. Extreme learning machines. IEEE Intell. Syst. 28, 30–59 (2013).

26. Bucurica, M., Dogaru, R. & Dogaru, I. A comparison of Extreme Learning Machine and Support Vector Machine classifiers. in Proceedings - 2015 IEEE 11th International Conference on Intelligent Computer Communication and Processing, ICCP 2015 471–474 (2015). doi:10.1109/ICCP.2015.7312705.

27. Huang, G., Huang, G. Bin, Song, S. & You, K. Trends in extreme learning machines: A review. Neural Networks 61, 32–48 (2015).

28. Maimakov, T., Sadykova, L., Kalmataeva, Z., Kurakpaev, K. & Šmigelskas, K. A Fast and Accurate Online Sequential Learning Algorithm for Feedforward Networks. Med. 49, 335–340 (2013).

29. van Schaik, A. & Tapson, J. Online and adaptive pseudoinverse solutions for ELM weights. Neurocomputing 149, 233–238 (2015).
30. Roy, M., Bose, S. K., Kar, B., Gopalakrishnan, P. K. & Basu, A. A Stacked Autoencoder Neural Network based Automated Feature Extraction Method for Anomaly detection in On-line Condition Monitoring. in Proceedings of the 2018 IEEE Symposium Series on Computational Intelligence, SSCI 2018 1501–1507 (Institute of Electrical and Electronics Engineers Inc., 2019). doi:10.1109/SSCI.2018.8628810.

31. Ahmed, F. E. Artificial neural networks for diagnosis and survival prediction in colon cancer. Molecular Cancer vol. 4 29 (2005).

32. Zhang, Y. D., Pan, C., Chen, X. & Wang, F. Abnormal breast identification by nine-layer convolutional neural network with parametric rectified linear unit and rank-based stochastic pooling. J. Comput. Sci. 27, 57–68 (2018).

33. Munir, K., Elahi, H., Ayub, A., Frezza, F. & Rizzi, A. Cancer diagnosis using deep learning: A bibliographic review. Cancers (Basel). 11, (2019).

34. Shanthi, P. B., Faruqi, F., Hareesha, K. S. & Kudva, R. Deep Convolution Neural Network for malignancy detection and classification in microscopic uterine cervix cell images. Asian Pacific J. Cancer Prev. 20, 3447–3456 (2019).

35. Shen, L. et al. Deep Learning to Improve Breast Cancer Detection on Screening Mammography. Sci. Rep. 9, 1–12 (2019).

36. Kingma, D. P. & Ba, J. L. Adam: A method for stochastic optimization. in 3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings (International Conference on Learning Representations, ICLR, 2015).

37. Chiang, H. T. et al. Noise Reduction in ECG Signals Using Fully Convolutional Denoising Autoencoders. IEEE Access 7, 60806–60813 (2019).

38. Wang, Y. et al. Assessing Optimizer Impact on DNN Model Sensitivity to Adversarial Examples. IEEE Access 7, 152766–152776 (2019).

39. Zhang, L., Gao, H. J., Zhang, J. & Badami, B. Optimization of the Convolutional Neural Networks for Automatic Detection of Skin Cancer. Open Med. 15, 27–37 (2020).

40. Rączkowski, Ł., Możejko, M., Zambonelli, J. & Szczurek, E. ARA: accurate, reliable and active histopathological image classification framework with Bayesian deep learning. Sci. Rep. 9, 1–12 (2019).