Journal of Advances in Mathematics and Computer Science

23(1): 1-25, 2017; Article no.JAMCS.34357

Previously known as British Journal of Mathematics & Computer Science

ISSN: 2231-0851

Image-Based Identification of Cell Cultures by Machine Learning

Oluleye Babatunde[1], Ashley Baltes[1] and John Yin[1*]

[1] *Department of Chemical and Biological Engineering, Systems Biology Theme, Wisconsin Institute for Discovery, University of Wisconsin-Madison, Madison 53706, Wisconsin, USA.*

Authors' contributions

This work was carried out in collaboration between all authors. Authors OB and JY conceived and designed the experiment. Author AB designed culture device with clamp apparatus and provided images. Authors AB and JY contributed reagents and materials. Author OB designed all the algorithms and implemented the software. Authors OB, JY and AB wrote the paper. All authors read and approved the final manuscript.

Article Information

DOI: 10.9734/JAMCS/2017/34357

Editor(s):

(1) Vitor Carvalho, Polytechnic Institute of Cavado and Ave, Portuguese Catholic University and Lusiada University, Portugal.

Reviewers:

(1) Abdullah Sonmezoglu, Bozok University, Yozgat, Turkey.

(2) Nilesh Bhaskarrao Bahadure, Chhattisgarh Swami Vivekanand Technical University, Bhilai, India.

(3) S. Rohith, Visvesvaraya Technological University, India.

(4) Nallasivan Gomathinayagam, Manonmaniam Sundaranar University, India.

Complete Peer review History: http://www.sciencedomain.org/review-history/19742

Original Research Article

Received: 25th May 2017

Accepted: 20th June 2017

Published: 28th June 2017

Abstract

Biomedical laboratories often use different cell types in the same assay or the same cell type in different assays. One cell type can become contaminated by another, or cells can be mis-identified, giving poor results. Addressing these issues by DNA analyses can be time-consuming, labor intensive or costly to implement. Here we uniquely employ Legendre moments (LM), Zernike moments (ZM), circularity and a genetic algorithm (GA) to advance a computer-based vision system, and we task it to identify four cell types used in virology: HeLa, Vero, BHK and PC3. By employing a k-nearest neighbor (kNN), multilayer perceptron (MLP), Convolutional Neural Networks (CNN) classifiers and a GA-selected 9-vector candidate comprising 4 ZMs, 4 LMs, and circularity, we provide adaptive system for deep machine learning. Our approach provides avenue to measure the performances of two of the conventional and popular classifiers (kNN and MLP) with a relatively recent classifier (CNN). We provide detailed mathematical treatments of the image signatures for accessibility and reproducibility in computer vision. Our methods are unique in biomedical applications. The performance of the kNN for k = 1, 2, and 3 using 10-fold cross-validation

**Corresponding author: E-mail: john.yin@wisc.edu, hezecomp@yahoo.com;*

yielded accuracies of (83.59%, 82.03%, 81.25%) and (84.38%, 82.82%, 82.03%) for 8-class and 4-class training sets, respectively, drawn from the same data while those of the MLP and CNN were 86% and 87.25% respectively. These results establish the feasibility of reliable automated cell identification, with diverse applications in biological and biomedical research.

Keywords: Image analysis; machine learning; circularity; legendre moments; zernike moments; biomedical cell images of cells; virology.

2010 Mathematics Subject Classification: 53C25, 83C05, 57N16.

1 Introduction

Recent advances in the live microscopy and imaging, such as high-throughput single-cell dynamic imaging[1], super-resolution microscopy [2], and light-sheet-based fluorescence microscopy [3], have enabled the capture of live-cell, tissue and whole-organism images with unprecedented spatial and temporal resolution. These tools and approaches have contributed to a deeper basic understanding of cell structure and function, organismal development, and microbiology, including virus-cell interactions. Although new insights have been chiefly based on human visual interpretation of images, the rapidly expanding volume of imaging data creates a pressing need for automation, employing computer-based vision systems that can do quantitative image analyses. To date, automated image-based analysis of cells, or image cytometry [4], has been primarily developed and implemented to identify cells and cell types ([5], [6], [7], [8], [9], [10]). For example, the passage of light through a cell-sized object and diffuser was captured by a camera to produce its opto-biological signature (OBS) in [8]. The OBS value of an object depends on optical characteristics of the object, including its morphology, size, and index of refraction. Distributions of OBS values were used to determine mean, variance, skewness, kurtosis, and entropy for different microscopic objects, which were then used to identify cells using a random forest classfier. In another work by [9], blood cells were identified using global pattern averaging for feature extraction and classification by an artificial neural network (ANN). To diagnose lung cancer an ANN based on a two-level ensemble architecture was used to classify cell images prepared from patient biopsies [11]. In this work level 1 categorized cells into binary outputs viz normal or cancerous, and level 2 further classified the cancerous cells as adeno-, squamous , small or large cell carcinomas. Despite these advances, images of cells are often subjected to translation, rotation or scaling [12], geometric transformations that adversely impact classifier performance [13]. There have been recent works (such as [14], [15]) that address such problems but down-to-earth implementation and description of the underlying algorithms (which of course will result in reproducible research), are lacking. Further we provide a genetic algorithm (GA) to select subset of the original feature space for improving the accuracy of the underlying machine learning model. From mathematical point of view, we provide the detailed treatments and derivation of ZM and LM. A combination of a GA, amalgation of LM, ZM, and circularity and detailed descrition of all the methods employed are considered to be valuable and novel in this field.

Here we address these challenges by incorporating circularity, Legendre moments (LM), and Zernike moments (ZM) into a k-nearest neighbor (kNN), MLP, and CNN classifiers, providing an adaptive computer-based vision system for cell type identification. Moreover, we detail and demonstrate reproducible features of the 9-vector candidate used in the training set, elucidating how LM and ZM contribute to the classifier performance. We hope that inclusion of such details will foster reproducible advances in computed-based vision systems for cell identification.

This paper is developed in several sections. In the next section we review the cell types widely used in the study of virus-cell interactions, and we follow with materials and methods for cell culture and image acquisition. Then we present the image descriptors used in this work, followed by our experimental design steps: a summary of the image dataset, image pre-processing, segmentation, feature extraction, feature selection, and classification. Finally, we discuss our results and indicate future directions for this research.

2 Common Cell Types Used in Virology

In order to grow and study viruses in the laboratory, one needs to maintain live cell cultures, which serve as hosts for virus infection. Cells that can be cultured indefinitely have typically been derived from tumors or cancerous growths, which have lost or disabled their control of proliferation. Different culturable cell types are susceptible to infection by different viruses. Here we provide an overview of four cell types commonly used in virology: HeLa, Vero, BHK and PC3. Detailed information on these and many other cell types are available from the American Tissue Culture Collection (ATCC) (https://www.atcc.org).

2.1 HeLa cells

HeLa cells are among the oldest and most widely used in scientific research, including virology ([16], [17]). The HeLa cell line was derived from cervical cancer cells taken from Henrietta Lacks an African-American patient who died of cancer in 1951 at the age of 31. Cells from her tumor were taken without her consent by scientist George Gey, who later discovered that the cells could be kept alive if spent growth medium were replenished. The remarkable history of Henrietta Lacks and the HeLa cell line have the subject of a best-selling book [18]. In addition to their widespread use in biomedical research, HeLa cells have also been used extensively to test for potential human sensitivity to tape, glue, cosmetics and many other consumer products ([19], [20]).

2.2 Vero cells

Vero cells were derived from kidney epithelial tissues extracted from an African green monkey. The 'Vero' lineage was originally developed in 1962 by Yasumura and Kawakita at Chiba University in Japan [21], and it was named as an abbreviation of *verda reno*, meaning "green kidney" in the Esperanto dialect. The cells have many uses such as screening for bacterial toxins, serving as host cells for the culture of viruses (e.g., polio, measles, influenza) and viral vaccine production, as well as for the culture of parasites such as trypanosomatids. The Vero cell lineage can be replicated through many cycles of division without becoming senescent. The cells have an abnormal number of chromosomes, a condition known as aneuploidy[22].

2.3 BHK cells

Baby hamster kidney (BHK or BHK-21) fibroblasts are an adherent cell line widely used in molecular biology. The cell line was derived in 1961 from the kidney of a baby Syrian hamster. BHK cells are susceptible to infection by many viruses, including human adenovirus D, reovirus 3, and vesicular stomatitis virus. This cell line has been utilized as a host for transformation with expression vectors containing selectable and amplifiable marker DNAs. They lack an intact innate immune response, so virus infections of BHK cells can produce robust yields of progeny virus [23].

2.4 PC3 cells

Human prostate cancer (PC3) cells are widely used in prostate cancer research. They were established in 1979 from the bone metastasis of prostate cancer in a 62-year-old Caucasian male. They have been particularly valuable in biomedical research to investigate biochemical changes in advanced prostatic cancer cells and in drug testing for potential therapeutic treatments ([24], [25], [26], [27]). Further, PC3 cells are being used to create human tumors in mice, which provide a model for the human tumor development within the context of a living host [28]. Finally, PC3 cells maintain an active innate immune response, making them useful to study how cellular immune signaling activates to suppress viral growth and spread ([29], [30]).

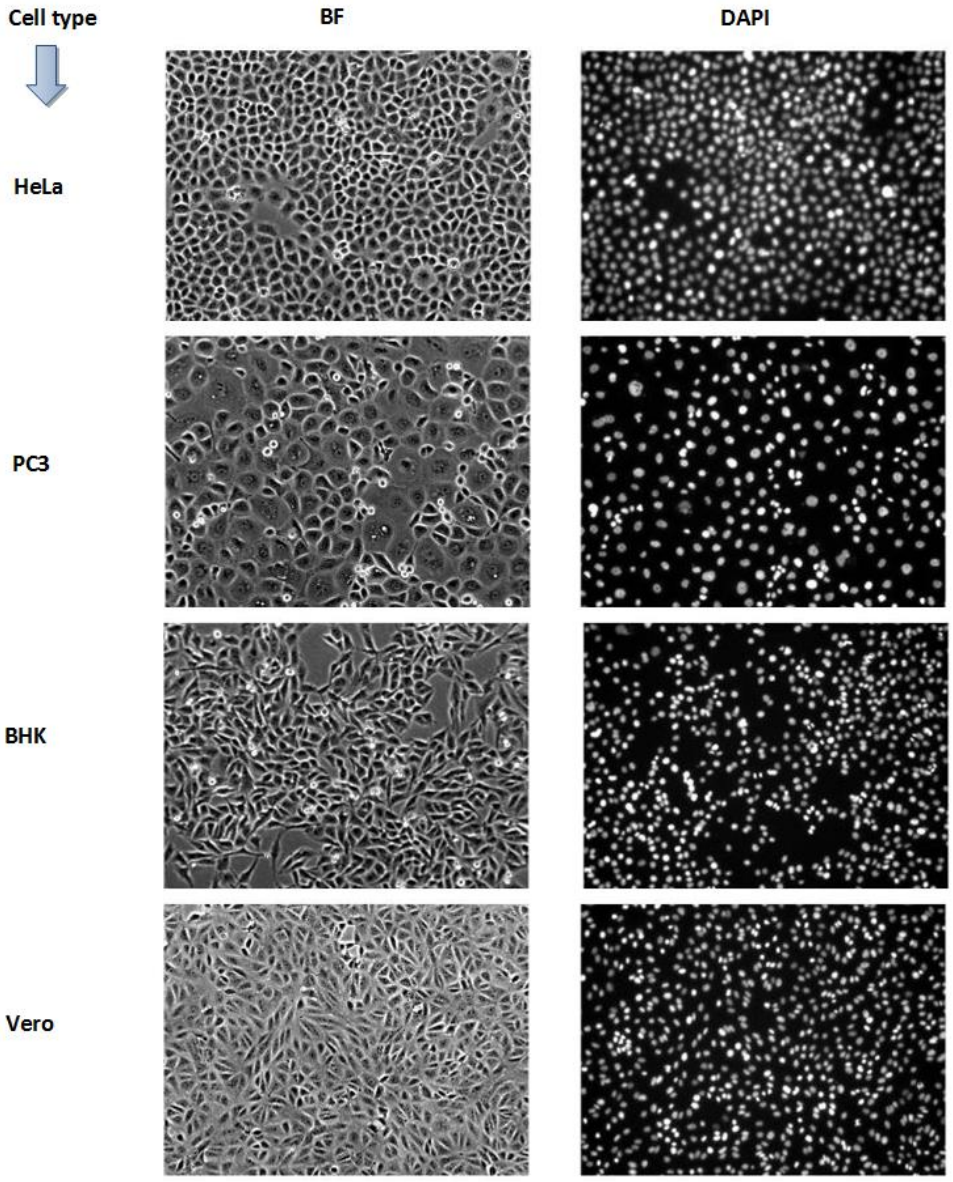


Fig. 1. A sample of both BF and DAPI versions for HeLa, PC3, BHK, & Vero cells

3 Materials and Methods (Virology Laboratory)

3.1 Cell culture

PC3 cells (described above) were obtained from ATCC (Manassas, VA, CRL-1435) and cultured in RPMI 1640 (Gibco) with 10% fetal bovine serum (FBS, Atlanta Biologicals). Baby hamster kidney (BHK-21), HeLa

(ATCC CRL-1958), and Vero (ATCC CCL-81) cells were cultured in Eagle's minimum essential medium (EMEM, CellGro) with 10% FBS and 2mM Glutamax I (Gibco) at 37° and 5% CO_2. Cells were stained with Hoechst 33342 (AnaSpec Inc. Cat 83218) diluted 1:20,000 in 2% FBS MEM for 1 hour before imaging.

3.2 Imaging

Imaging was done on a Nikon Eclipse TE300 inverted microscope equipped with an EXi Aqua CCD camera (QImaging, Surrey, BC, Canada). Illumination for imaging the Hoechst stain was provided by a PhotoFluor (Chroma, Bellows Falls, VT): the excitation filter, emission filter, and multichroic mirror used in this work were from the Sedat Quad set 86000v2 (Chroma). Illumination for the phase contrast images was provided by a Nikon TE-PS100. Images were obtained through a Nikon Plan Fluor 10x, 0.30 NA Objective. A digital camera attached to a microscope enables the division of the field view into a grid of pixels. The intensity of the light absorbed is numerically quantified as pixel values. Computers work with these digital images which themselves, are grid of numbers indicating the luminous intensity. Image analysis are purposely applied to these images to transform them into measurements of biological relevance such as cardinality of cells, the texture and color of cells, etc, in a given image.

4 Image Descriptors

4.1 Circularity

Circularity was used as one of the image signatures in this work. Circularity or compactness is defined as a measure of similarity between a 2D shape and a circle. In other words, circularity defines the degree to which a shape differs from a perfect circle. It is the ratio between the quantity $4\pi a$ (a being the area of the cell's image) and the square of its perimeter p . It is given as

$$Circularity = \frac{4\pi a}{p^2} = \frac{p^2}{4\pi a} \tag{4.1}$$

The circularity of a circle is 1 (i.e $\frac{4\pi a}{p^2} = \frac{4\pi.\pi r^2}{(2\pi r)^2} = 1$). The comparision of a given shape with a circle is a measure of both compactness or circularity. The compactness (circularity) is not only invariant to translation and rotation but also to scaling [31]. Many cells look circular in nature and thus, the image of such could be described by circularity.

4.2 Image moments

Other image signatures used in this work are image moments which are Zernike moments (ZM) and Legendre moments(LM). Image moments are global region-based descriptors for shape and is synonymous to combination of area, compactness, irregularity, and higher order descriptors together [13]. Moments are good image descriptors for characterizing smoothness and regularity of images. The digital images used in this study could be defined as a 2D light intensity function , say, $f(x,y)$, where x and y are the spatial coordinates of the image and f denotes the brightness or gray level at the point (x,y). If the image is generated from a physical scenario, then the intensity values of such image are proportional to energy radiated by the physical source. Therefore, $f(x,y)$ is always assumed to be nonzero and finite as given by the inequality $0 < f(x,y) < \infty$. An image moment of microscopic cell is defined as the integration of an image function, say, $f(x,y)$, with a region-defined polynomial basis ([32], [33]). The region here is defined as the area where that image is valid. Example of such region is a 2D cartesian plane or image xy-space. From ([34], [35]), the general moment M_{pq} of any image $f(x,y)$ of order $p+q$, where $p > 0, q > 0$, is defined as:

$$M_{pq} = \int \int_D pol_{pq}(x,y) f(x,y) dxdy \tag{4.2}$$

where $pol_{pq}(x,y)$, $i = 1(1)p$, $j = 1(1)q$ are polynomials basis functions defined on domain D.

4.2.1 Legendre moments

Legendre moments (LMs) are based on the Legendre polynomial (LP) ([12], [36], [37]). The LPs themselves are the everywhere regular solutions of Legendre's equation (Eq 4.3),

$$(1-x^2)u'' - 2xu' + n(n+1)u = 0, n = 0, 1, 2, ... \tag{4.3}$$

which is a class of 2nd order linear ordinary differential equation (ODE).

The nth order LP is defined by

$$P_n(x) = \sum_{m=0}^{K} a_{n-2m}x^{n-2m} = \sum_{m=0}^{K} (-1)^m \frac{(2n-2m)!}{2^n m!(n-m)!(n-2m)!}x^{n-2m}, \quad K = \frac{n}{2} \text{ or } K = \frac{n-1}{2} \tag{4.4}$$

The $(m+n)th$ order of Legendre moment for a given image of intensity $f(x,y)$ defined on the square $[-1,1] \times [-1,1]$ is

$$L_{m,n} = L_{mn} = \frac{(2m+1)(2n+1)}{4} \int_{-1}^{+1}\int_{-1}^{+1} P_m(x)P_n(y)f(x,y)dxdy \tag{4.5}$$

where $m,n = 0, 1, 2, ...$

The microscopic images used were all rectangular. The LM for digital images in a square domain $N \times N$ are given as Eq 4.6. The images were all scaled to be in the region $-1 \leq x,y \leq 1$. As the order of the polynomial increases, the precision required to describe the given object also increases. The LM can easily be computed from the conventional moments and the well-defined polynomial coefficients. Teague has demonstrated that higher order moments (greater than three) have significant information and may be necessary to usefully characterize an image for a given application [38].

$$L_{m,n} = L_{mn} = \frac{(2m+1)(2n+1)}{(N-1)^2} \sum_{i=1}^{N}\sum_{j=1}^{N} P_m(x_i)P_n(y_j)f(x_i,y_j)dxdy \tag{4.6}$$

where

$$x_i = \frac{2i-N-1}{N-1} \tag{4.7}$$

and

$$y_j = \frac{2j-N-1}{N-1} \tag{4.8}$$

Using the orthogonality property of LM, a given image can be reconstructed from a finite number of moments of order ranging up to (N,N) and this is expressible as Eq 4.9.

$$\bar{f}(x,y) \approx \sum_{i=0}^{N}\sum_{j=0}^{N} P_i(x)P_j(y)L_{ij} \tag{4.9}$$

All the ideas adopted from section 4.2.1 have been put together as a complete algorithm (Algorithm 1) which will be used later in feature extraction (see section 5.4).

4.2.2 Zernike moment

The Zernike moments(ZM) have been defined as a set of complete complex orthogonal basis functions that are square integrable and defined over the unit disk (see Fig. 2). An open disk around a given point, say, x in a

Algorithm 1 Computation of Legendre Moments

procedure COMPUTELEGENDREMOMENTS()
  Input image $f(x_i, y_j)$, $i = 1(1)M$, $j = 1(1)N$
  Normalize $f(x,y)$
  Compute $\Delta x_i = \frac{2}{M}, \Delta y_j = \frac{2}{N}$
  for $i = 1(1)M$
    for $j = 1(1)N$
$x_i{}^* \mapsto -1 + (i - \frac{1}{2})\Delta x_i$
$y_j{}^* \mapsto -1 + (j - \frac{1}{2})\Delta y_j$
$L_{m,n} \mapsto \frac{(2m+1)(2n+1)}{4} \int_{-1}^{+1}\int_{-1}^{+1} P_m(x_i{}^*)P_n(y_j{}^*)f(x_i{}^*, y_j{}^*)dxdy$
  Endfor.
    Endfor
  Output $L_{m,n}$
end procedure

plane is the set of all points in the plane whose distance from x is less than 1 (see Eq 4.10). For a closed disk, the distance from x is less than or equal to ρ where ($\rho = 1$ as shown in Eq 4.11).

$$D(x) = \{y \in \mathbb{R} : \|x - y\| < \rho\} \tag{4.10}$$

$$\bar{D}(x) = \{y \in \mathbb{R} : \|x - y\| \leq \rho\} \tag{4.11}$$

Radial moments are general defined on a closed disks and as such, there is a strong connections between radial Zernike moments and radial moments. The radial moments of order p with repetition q are defined as:

$$D_{pq} = \int_{\theta=0}^{2\pi}\int_{r=0}^{\infty} r^p e^{-iq\theta} f(r,\theta) r dr d\theta, \quad i = \sqrt{-1}, \quad p = 0,1,2,...,\infty \ \ and \ \ q \in Z^+. \tag{4.12}$$

ZM are orthogonal moment based on Zernike polynomials. Orthogonality here implies that there is no redundancy or overlapping of information between the moments. Thus moments are uniquely quantified based on their orders ([34], [39]). The distinguishing feature of ZM is the invariance of its magnitude with respect to rotation. If we are given the ordered pair (m,n) which represents the order of the Zernike polynomial and the multiplicity of its phase angle, then the Zernike moment, Z_{nm} for any given sample image $\{f(x_i, y_j) : 1 \leq i \leq M, 1 \leq j \leq N\}$, can be calculated as Equations (4.13) or (4.14)

$$Z_{nm} = \frac{n+1}{\pi} \int\int_D f(x,y) V_{nm}^*(x,y) dxdy = \frac{n+1}{\pi} \sum_x^M \sum_y^N V_{nm}^*(x,y) f(x,y) \tag{4.13}$$

where $x^2 + y^2 \leq 1$, and $m = 0,1,2,3,...\infty$. The m defines the order of the Zernike Polynomial while n which is either negative or positive, represents the multiplicity of the phase angles in ZM.

$$Z_{nm} = \frac{n+1}{\pi} \int_0^{2\pi}\int_0^1 f(\rho,\theta) R_{nm}(\rho) e^{-im\theta} \rho d\rho d\theta \tag{4.14}$$

$$V_{nm}(\rho,\theta) = R_{nm}(\rho) e^{im\theta}, \theta \leq 1 \tag{4.15}$$

$$V_{nm}(\rho,\theta) = R_{nm}(\rho) e^{im\theta}, \theta \leq 1 \tag{4.16}$$

where

$$\rho = \sqrt{x^2 + y^2}, \theta = \arctan(\frac{y}{x}) \tag{4.17}$$

are the image pixel radial vector and θ is the angle between it and x-axis respectively

$$R_{nm}(\rho) = \sum_{a=0}^{\frac{(n-|m|)}{2}} (-1)^a \frac{(n-a)!}{a!(\frac{(n+|m|)}{2}-a)!(\frac{(n-|m|)}{2}-a)!} \rho^{n-2a} \tag{4.18}$$

ZM are related to radial moments shown in Eq 4.12 as :

$$Z_{pq} = \lambda_p \sum_{k=q}^{p} R_{pq} D_{pq}, \quad \lambda_p = \frac{p+1}{\pi} \tag{4.19}$$

The R_{nm} is the Zernike radial basis polynomial. The following conditions must be satisfied:

$$(a) n \in Z^+$$

$$(b) n - |m| \quad is \ \ even$$

$$(c) |m| \leq n$$

$$(d) \int_0^{2\pi} \int_0^1 V_{nm}^*(\rho,\theta)\rho d\rho d\theta = \frac{\pi}{n+1} \delta_{np}\delta_{mq}, \delta_{zv} = \begin{cases} 1 & z = v, \\ 0, & otherwise \end{cases} \tag{4.20}$$

In a more compact form, Zernike basis functions are defined with an order m and a repetition n over $D = \{(m,n)|\ 0 \leq m \leq \infty,\ |n| \leq m,\ |m-n| = even\}$ and a notable numerical property of Zernike polynomial is that they are always in the range -1 to $+1$ as given in the following expression:

$$|Z_n^m(\rho,\theta)| = |Z_{nm}(\rho,\theta)| \leq 1 \tag{4.21}$$

A notable fact to also know is that one ZM is a complex number that contains two different values: *magnitude or amplitude* and *phase angle* but the proper way of applying ZM is to use the magnitude as it is inherently invariant to rotation.

The ROI is mapped to the unit disc (using Eq 4.17) through polar coordinates, where the center of the ROI is the origin of the unit disk. The conversion from rectangular to polar coordinates is done through Eq (4.17). The coordinates are then described by the length of the vector from the origin of the disk to the coordinate point ρ , and the angle from the x-axis, to the vector ρ, (the polar radius). The polar angle is represented as θ. The pixels falling outside the unit disc are not used in the calculation. The translation invariance is achieved by moving the centroid of the ROI to the origin of the disk and this eventually causes $m_{01} = m_{10} = 0$. The centroid of the ROI is given by the coordinates $(\bar{x},\bar{y})$ where

$$\bar{x} = \frac{m_{10}}{m_{00}}, \bar{y} = \frac{m_{01}}{m_{00}} \tag{4.22}$$

The scale invariance for ZM is achieved through normalization of the image so that the total area of the foreground pixels is of predetermined value, say, β. If the scaled version of the image $f(x,y)$ is represented as $f(x/\alpha, y/\alpha)$, the regular moment m_{pq} of $f(x,y)$ and m_{pq}^1 of $f(x/\alpha, y/\alpha)$ are related by:

$$m_{pq}^1 = \int_x \int_y x^p y^q f(\frac{x}{\alpha}, \frac{y}{\alpha}) dxdy \tag{4.23}$$

$$= \int_x \int_y \alpha^{p+q+2} x^p y^q f(x,y) \alpha^2 dxdy \tag{4.24}$$

$$= \alpha^{p+q+2} \int_x \int_y f(x,y) dxdy \tag{4.25}$$

$$= \alpha^{p+q+2} m_{pq} \tag{4.26}$$

The aim here is to have $m_{00}^1 = \beta$ and so, let

$$\alpha = \sqrt{\frac{\beta}{m_{00}}} \tag{4.27}$$

and then substitute for α in m_{00}^1 to have $m_{00}^1 = \alpha^2 m_{00} = \beta$. Therefore, translation and scaling invariance is achieved through the formula in Equation 4.28.

$$g(x,y) = f\big(\frac{x}{\alpha} + \bar{x}, \frac{y}{\alpha} + \bar{y}\big) \tag{4.28}$$

where

$$\alpha = \sqrt{\frac{\beta}{m_{00}}} \tag{4.29}$$

Algorithm 2 shows the pseudocode adopted for computing the ZM. The pseudocode was based on all the equations in section 4.2.2.

In a similar way to LM in section 4.2.1, the reconstruction of the pattern or image can be expressed as the sum of every Zernike basis functions weighted by the corresponding moments (of order, say, N):

$$\bar{f}(x,y) = \sum_{(i,j=0)}^{N} \sum_{(i,j=0)}^{N} Z_{ij} V_{ij}(x,y) \tag{4.30}$$

where Z_{ij} and V_{ij} are as given in Algorithm 2.

Algorithm 2 Computation of Zernike Moments

```
procedure COMPUTEZERNIKEMOMENTS()
    Input N_max, image f(x_i, y_j), ,i = 1(1)M, j = 1(1)N
    Normalize f(x,y)
    Z_nm = 0
    for x = 0 to 1
            for y = 0 to x
                    for n = 0 to N_max
    Compute [R_n0, R_n1, ..., R_nm] using Equation 4.18
          Endfor
              Endfor
                  Endfor
    for m = 0 to n
    Compute V(ρ(x,y), θ) using Equation 4.16.
          Z_nm = Z_nm + R_nm * V(ρ(x,y), θ)
       Endfor
    Output Z_nm
end procedure
```

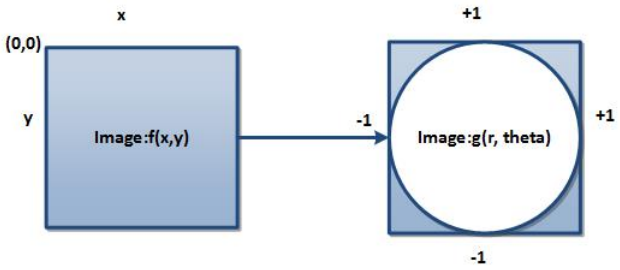


Fig. 2. Conversion from rectangular to polar coordinates

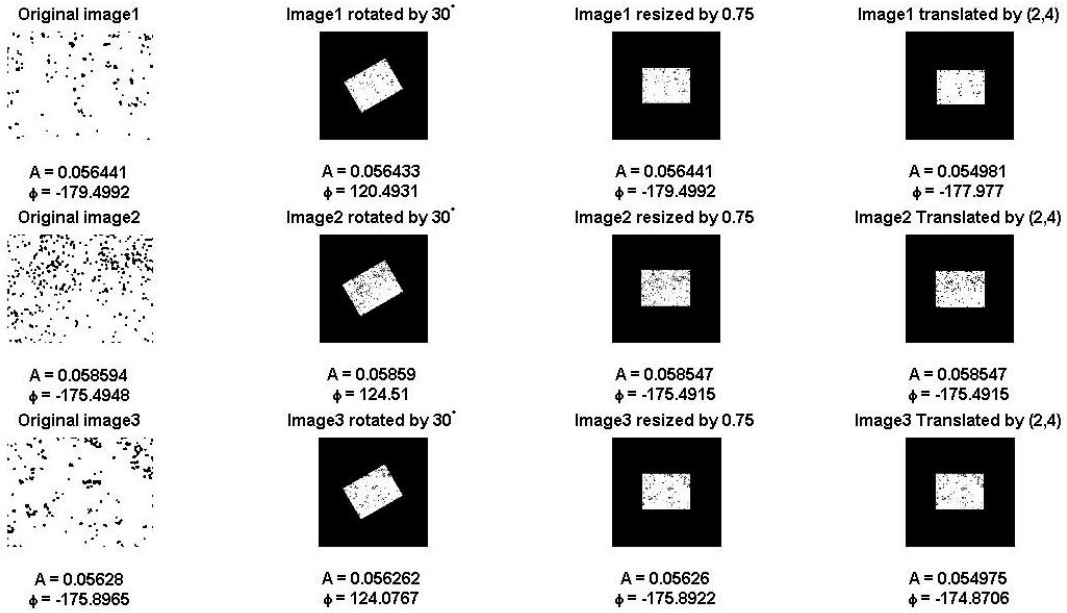


Fig. 3. Invariant property of ZM. Here, set of three images of HeLa, BHK and PC3 respectively were subjected to rotation, scaling and translation. The ZM descriptors extracted from these images (represented as A) are appromixately the same, showing invariant property of ZM- a highly desirable factor in machine learning or pattern recognition

5 Design of Computer-based Vision System

The description of the computer-based vision system developed in this work is diagramatized in Fig.4. The choice of classification model is the kNN. The kNN was chosen because it is both simple and easy to implement. All algorithms in this work were implemented completely in MATLAB programming language. The performance of the kNN is benchmarked with two other (standard) classifiers.

5.1 Image dataset

The dataset in this work comprised of microscopic images of the HeLa, PC3, BHK and Vero cells. These four cells were plated at 4 densities 1, 2, 3, 4 (1-high, 4-low) and stained with Hoechst. Bright Field (BF) and DAPI

images were taken. A DAPI (4′, 6-diamidino-2-phenylindole) is a fluorescent stain that binds strongly to A-T rich regions in DNA. It is used extensively in fluorescence microscopy. It can be used to stain live and fixed cells because it can pass through an intact cell membrane [40]. A Bright-field (BF) microscopy is the simplest of all the optical microscopy illumination techniques. A sample illumination in BF microscopy is illuminated from below and observed from above white light. The contrast in the sample is caused by absorbance of some of the transmitted light in dense areas of the sample.The name BF is adopted from the typical appearance of a bright-field microscopy image which looks like a dark sample on a bright background ([41], [42]). There were 40 replicates (BF and DAPI) acquired for each cell type and density combination for a total samples with 1280 images. The images are stored in folders labeled with the cell type and subfolders that identify the density.

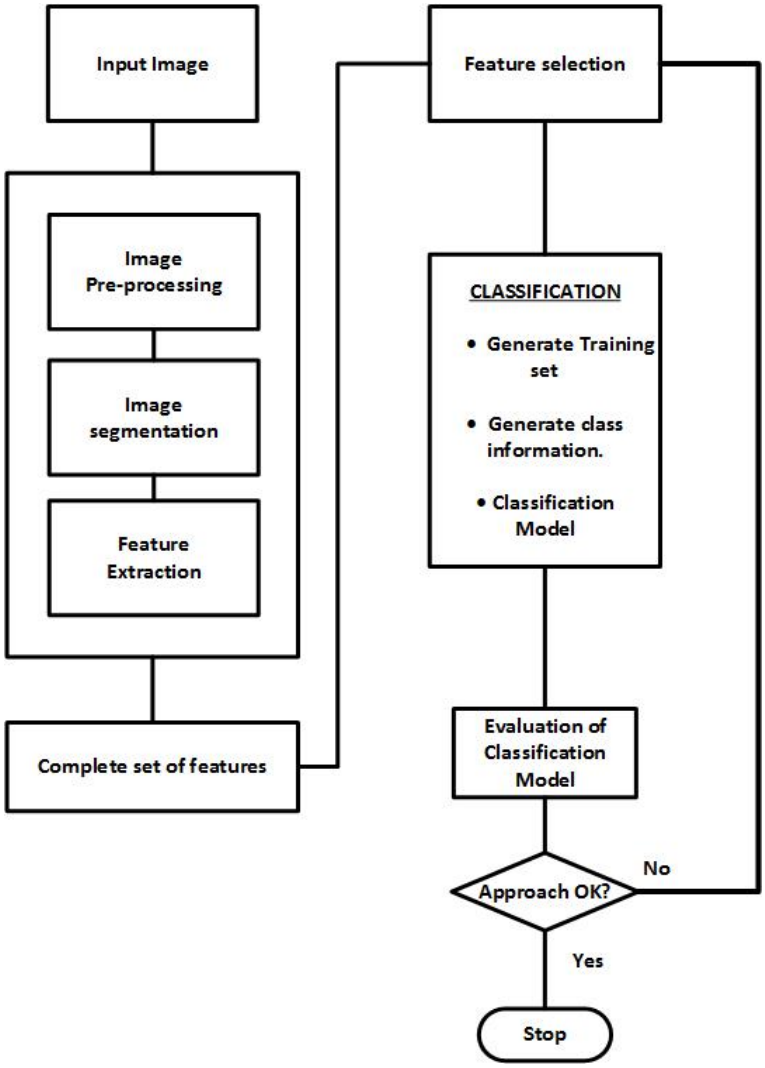


Fig. 4. Research Methodology for image-based identification of living cells [43]

5.2 Image pre-processing

The images in the dataset were all in .tiff format and each with dimension 1392×1040 pixesl having bit depth of 96 dpi for both horizontal and vertical resolution. The size of each image was 2.77 MB. The original images could not be seen with ordinary image display function. The entire images in the dataset were all thus, subjected to contrast adjustment before feature extraction and before each could be seen properly.

5.3 Image segmentation

After the pre-processing steps above the entire images in the dataset were further binarized and segmented to aid the next step wich is to extract required features from the images. Otsu thresholding was used to separate the foreground pixels from the background pixels. The image segmentation made the feature extraction easier to carry out since the images were then in the required format needed for the computation of the image descriptors (or image features) which are circularity, LM, and ZM.

5.4 Feature extraction

The image descriptors employed in this work are circularity, Legendre Moments (LM), and Zernike Moments (ZM). The rational for combining such features is due to the simplity and effectiveness of the circularity and the affine nature of both LM and ZM which have been established in some literatures [34]. The LM and ZM were extracted using Algorithms 1 & 2 while the circularity was extracted using Equation 4.1. The selected features are shown in Fig. 6. The figure shows the 9-vector candidate in each row which represents circularity, 4 LM and 4 ZM extracted from each image in the dataset.

5.5 Feature selection

Feature selection is often needed in building machine model as all the available features might not contribute to the performance (accuracy improvement) of the system. Thus, feature selection is done prior to classification. Feature subset selection (FSS) is an operator Fs or a map from an m-dimensional feature space (input space) to n-dimensional feature space (output) given in mapping,

$$F_s : \mathbf{R}^{r\times m} \mapsto \mathbf{R}^{r\times n} \tag{5.1}$$

where $m \geq n$ and $m,n \in \mathbf{Z}^+$, $\mathbf{R}^{r\times m}$ is any database or matrix containing the original feature set having r instances or observation, $\mathbf{R}^{r\times n}$ is the reduced feature set containing r observations in the subset selection. A GA was employed to select the best features from the original feature set. GA iteratively employ the use of one population of chromosomes (solution candidates) to get a new population using a method of natural selection combined with genetic functionals such as crossover and mutation (immitating Charles Darwin evolution principles of reproduction, genetic recombination, and the **survival of the fittest** ([34]). The parameters associated with the GA here are shown in Fig. 5a. Each chromosome (shown in Fig. 5b) is a binary string with gene value **'1'** indicates the particular feature indexed by the position of the **'1'** is selected. If it is **'0'**, the feature is not selected for evaluation of the chromosome. The chromosomes are the encoded bit strings represeting the features. As the GA iterates, the chromosomes in the current population are evaluated and ranked according to their fitnessess from the kNN-based classification error. As shown in Fig. 3, the best chromosome reports the best feature set.

$$Fitness = \frac{\alpha}{N_o - N_s} \tag{5.2}$$

where

1. α = kNN error.
2. N_o = cardinality of the original feature set

3. N_s = cardinality of the selected features.

The expression $LM(n_x, n_y)$ is assumed to represent Legendre moment of orders n_x and n_y in x and y spatial components of the images while the expression $ZM(m,n)$ is assumed to represent Zernike moment of polynomial order m and phase angle multiplicity n. The original features or image descriptors (signatures) extracted from the image dataset were all 81 in numbers. The GA reduced this to a nine-vector features were {circularity, $LM(2,1), LM(2,3), LM(3,2), LM(4,2), ZM(2,-2), ZM(4,2),$
$ZM(5,3), ZM(6,4)$}.

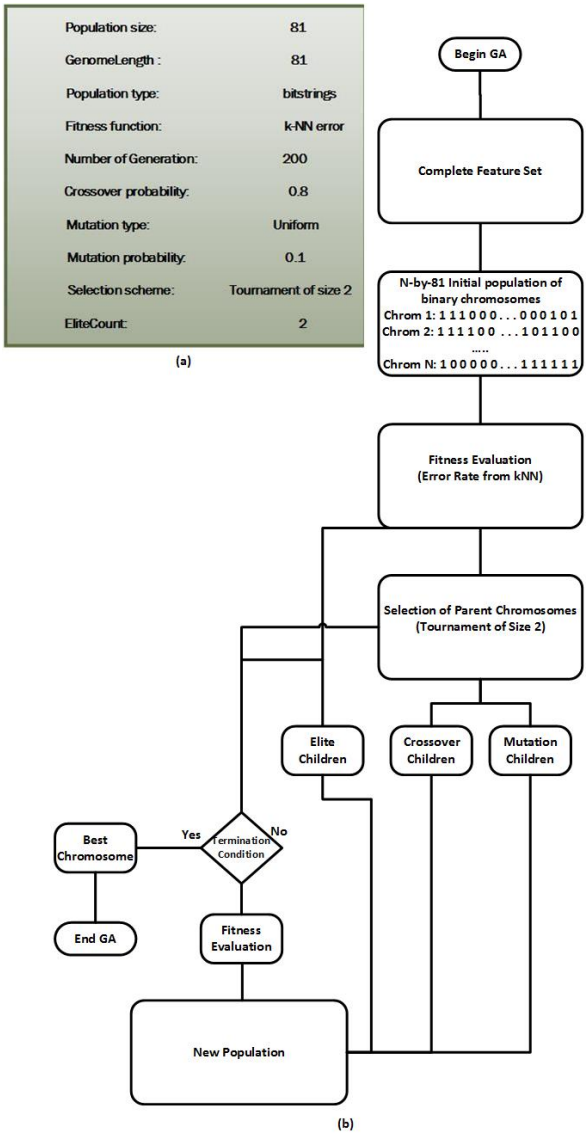


Fig. 5. GA Parameters and Flowchart [43]

Features

Cells

	F1	F2	F3	F4	F5	F6	F7	F8	F9
1	0.2608	0.1074	0.0245	0.1174	0.4471	0.0382	0.0078	0.0124	0.0167
2	0.2420	0.1278	0.0294	0.1121	0.4264	0.0417	0.0069	0.0133	0.0171
3	0.2877	0.0975	0.0218	0.1342	0.4653	0.0336	0.0074	0.0127	0.0169
4	0.2614	0.1250	0.0294	0.1462	0.4155	0.0401	0.0067	0.0113	0.0147

Fig. 6. Mean of circularity, Legendre Moments (LM) & Zernike Moments (ZM) across the cell classes. The columns represent the nine-feature vector extracted from the images in the dataset. The numbers $1,2,3,4$ in the figure represent the four classes of cells used (viz class1 (HeLa), class2 (PC3), class3 (BHK), class4 (Vero)). The column names $F1$ through $F9$ represents the mean of circularity, LM and ZM. The mean values for the extracted features across all the classes are different showing their applicability for classification purpose

5.6 Classification model

5.6.1 k Nearest neighbor

The kNN algorithm is useful for classification not requiring model building, and hence, it is called "instance-based learning". It solves the classification problem by looking for the shortest distance between the test data and training sets in the feature space [44]. The distance is generally computed in a Pythagorean sense (by finding the square root of the sum of differences) and that is what was adopted in this work. We represent our training set as x below:

$$\mathbf{X} = \{x_{ij}\} \tag{5.3}$$

Each x_{ij} (i = 1 (1) 1280, j = 1(1)9) is a scalar containing a feature extracted from each of the images. The kNN algorithm computes Euclidean distance between the test data x_{test} and the training set and then find the nearest point (shortest distance) from the training set to the test set as:

$$D(x_{test},x) = \sqrt{\sum_{j=1}^{M}(x_{test}-x_{ij})^2} \tag{5.4}$$

As usual the kNN classifier considers only the k nearest neighbors (local information) denoted by $x_{ij_1},...,x_{ij_k}$ as the member(s) of the set (a normed linear space).
A normed linear space, say, kNNSpace $(E, \| \, . \, \|)$ is a metric space that is endowed with the following properties:

1. $\| \, x \, \| \geq 0$ for all $x \in E$ and $\| \, x \, \| = 0$ if and only if $x = 0$
2. $\| \, \alpha x \, \| = |\alpha| \, \| \, x \, \|$ for all $x \in E$ and $\alpha \in \mathbb{R}$.
3. $\| \, x+y \, \| \leq \| \, x \, \| + \| \, y \, \|$ for all $x \in E$ (triangular inequality)

$$kNNSpace = \{x_j | d(x_{test},x_{ij}) \leq d(x,x_i)\} \tag{5.5}$$

The kNN rules involve classifying a test sample, say, x_{test} by assigning it the most frequently represented among the k nearest samples.

Since $k = 1,2,3$ in this research, the kNN count each category m in the class information (accumulated as $count(x_m)$) using $k = 1,2,3$ Nearest Neighbors and then report classification results based on the expression

$$argmax(count(x_m)) \tag{5.6}$$

subject to

$$\sum_{i=1}^{M} count(x_m) = class \tag{5.7}$$

where $class = \{1(1)N_c\}$, with $N_c \in Z^+$, the number of classes in the training set. Algorithm 3 summarizes the operation of the KNN used. The classification accuracy of the kNN algorithm is sensitive to the value of k (see Figs. 8,9,10,11,12,13 section 6).

Algorithm 3 Algorithm for k Nearest Neighbor

1: **procedure** COMPUTEKNN()
2: **Input** $TRAININGSET = \{x_i, c_i\}$, $x =$ feature set, $i = 1(1)M$, M = number of observations in the training set, c = class information, $j = 1(2)N_c$, N_c = number of classess available and test image x_{test}.
3: **Assign** $p_i \leftarrow \{x_i, c_i\}, i = 1(1)M, c_i \in N_c$ where $p_i =$ posteriories of x_i.
4: **Compute** $D(x_{test}, x_i) = \sqrt{\sum_{m=1}^{M} (x_{test} - x_{im})^2}$
5: **sort** p_i based on D.
6: **Select** the first k points from the sorted list
7: **Assign** ClassLabel $\leftarrow p^*$ if $c^* = argmax(count(x_i))$
8: **end procedure**

5.6.2 Multilayer perceptron (MLP)

The second classifier employed in this work is MLP which is typical feed-forward artificial neural network model with 3 layers which are input, hidden, and output layers with multiple neurons. The MLP can be represented as Eq. 5.8.

$$y_i = f\left[\sum_{k=1}^{N} \omega_k g\left(\sum_{j=1}^{J} (\omega_j x_j + \phi_j)\right) + \varepsilon_k\right] \tag{5.8}$$

where N = Number of hidden-layer neurons, ω_j = synaptic weights connecting the input and hidden layer neurons, ω_k = weights connecting the biases in the hidden and output layers, while $f(.)$ and $g(.)$ are respectively linear and sigmoid functions [45]. Further details about MLP may be found in [46], [47], [48].

5.6.3 Convolutional neural metworks (CNN)

We also employed a relatively new classifier (a convolutional neural network(CNN or ConvNet)) which is suited for deep machine learning. CNN is a variation of MLP. In CNN the neuronal connectivity pattern is inpired by the organization of the animal visual cortex. CNN is made up of neurons that have weights and biases that are trainable. Each neuron receives inputs (most commonly images), performs a dot product and optionally follows it with a non-linear function [49], [50]. Mathematically, a convolution operation involves shift, multiplication and sum operations. CNN is alternatively known as shift invariant or space invariant ANN because of its weights architecture and translation invariance characteristics. The CNN architectures assumes that the input data are

images. The assumption makes the forward function more efficient to implement and vastly reduce the amount of parameters in the network [49]. The ConvNet employed in this work consists of multiple layers, such as convolutional layers, max-pooling or average-pooling layers, and fully-connected layers. The neurons in each layer of a CNN are arranged in a 3-D data structure, mapping a 3-D input (i.e rgb image) into a 3-D output. With biomedical cell images (HeLa, Vero, BHK and PC3), the first layer (input layer) holds the images as 3-D inputs, with the dimensions being height, width, and color channels of the image. The neurons in the first convolutional layer connect to the regions of the cell images and map them into a 3-D output. The hidden units (neurons) in each layer learn nonlinear combinations of the original inputs, which is called feature extration. The features learned from one layer are known as activations and fed as inputs for the next layer. Finally, the learned features become the inputs to the classifier or the regression function at the end of the network [51].

Algorithm 4 k-fold Cross Validation for Classifion Model

1: **procedure** KFOLDCLASSIFIER(DATASET, K)
2: **Input** DataSet and the class information.
3: Randomly partition DataSet into K folds (disjoint sets) using the class information such as $X = X_1 + X_2$, where $(X, c_i) \in \mathbb{R}^D \times \{1,2,3,...,4(8)\}$
4: **DO** counter $\mapsto$ counter + 1
5: Remove k and train Classifier using feature from all classes except class k
6: Use X_2 for validation and X_1 for training
7: Compute $Error_{kNN}$ on the validation set X_2 as

$$Error_{Classifier}(X) = Classifier(X_2)$$

j = number of datapoints in the partition k
8: **UNTIL** counter = K

$$CV_{Error} = \frac{1}{N}\sum_{j=1}^{N} Error_{Classifier}$$

9: **end procedure**

6 Results and Discussion

The work here was based on identification of four cells that are commonly employed in virology laboratries. These cells include BHK, PC3, HeLa and Vero cells. In this section, we present two metrics commonly used in evaluating machine learning models. We describe both confusion matrix and receiver operating characteristics. It should be noted that this work also tested the effect of having increased class information on the performance of the learning machine. As such we used two types of training sets. The first training set had 4 classes representing HeLa, BHK, PC3 and Vero cells, while the second training set had 8 classes representing HeLa (DAPI and BF version), BHK (DAPI and BF version), PC3 (DAPI and BF version) and Vero (DAPI and BF version). kFold CV (see 4 was employed to eliminate biasness in model prunning. The classification output (result image) for both were the same but there was a little increase in accuracy for 4-classes training set than 8-classes training set. We also benchmarked the kNN classifier in this work against two other models namely MLP and CNN. The results shows that CNN outperformed the other two classifiers. The kNN is also close in terms of the accuracy. Although, kNN ranked third (or least), we have been able to show that kNN could be made to perform excellently well with proper enviromental settings such as appropriate image signatures and manifold reduction techniques such as GA.

6.1 Confusion matrices

In this section, the performance of our classification model is analysed using confusion matrices (see Figs. 8,9,10,11,12,13). A confusion matrix is a tabular tool or matrix display of the instances from the training set that were correctly and incorrectly predicted by classifiers in machine learning. It can be represented as ConfuseMatrix $\in R^{c\times c}$, a square matrix whose (backward) diagonal elements depicts the actual number of classes that are rightfully classified and c is the number of classes in the dataset. A confusion matrix is also called contigency table or error matrix since its all about visualising the performance of the learning algorithm. A confusion matrix M_{ij} containing $c(i,j)$ shows overall classification with regards to the whole (original training set). The diagonal entries show a metric representing number of images in each class (i.e BHK, PC3, HeLa and vero cell) that were correctly classified. In other words, it shows the number of class i that were correctly classified as j. Using the numbers shown in Figs. 8,9,10,11,12,13, we define an entry in ConfuseMatrix as the number of observations of cell image of class c_i that the classifier (kNN) predicts to be of class c_j, where $i = j = 1(1)4,8$. The classification accuracies in this work were computed from the confusion matrix M based on the following formular:

$$Accuracy = \frac{trace(M)}{sum(M)} \tag{6.1}$$

where $trace(M)$ is the sum of all the entries in the backward diagonal of the matrix M and $sum(M)$ is the sum of all the elements or entries of the matrix M.

1. 8-Classes Training Set
   The confusion matrices for the 8-classes training set and kNN {k = 1, 2, 3} are shown in Figs. 8,9 and 10. The backward diagonal colored in pink shows the numbers {14, 11, 13, 15, 15, 11, 16, 12}, {14, 11, 12, 15, 15, 10, 14, 14} & {15, 12, 14, 15, 14, 9, 14, 11} for $\{C1, C2, C3, C4, C5, C6, C7, C8\}$ for the kNN Classifier {k = 1, 2, 3}. These diagonal entries are the correct classification (or the true positives). In Fig. 8, it's shown that the system predicted 1 $C5$ and 1 $C7$ for $C1$. So for 16 actual classes for $C1$, only 14 were predicted correctly. For testing this learning machine, only 18 instances of unknown sample of each classes were used. The classes $\{C_i, i = 1(1)8\}$ represents both DAPI and BF version of each of HeLa, PC3, BHK, and Vero cells. It's to be noted that both DAPI and BF images represents a single cell type. So a display showing HeLa cell (DAPI) and that showing HeLA cell (BF) are all showing the same cell type. A variation in the classification display is to show to the non-expert in microscopy imaging that the images (DAPI and BF) were taken under different illumination and condition. Another objective of including versions for DAPI and BF is to see the effect of increasing the number of class information upon the performance of the classifier. The classification accuracies for kNN, $\{k = 1,2,3\}$ were shown to be $83.59\%, 82.25\%, 82.03\%$. MLP and CNN reported accuracies of 86% and 87.25% respectively. These numbers are computable using the Formular 6.1.
2. 4-Classes Training Set
   The confusion matrices for the 4-classes training set and kNN {k = 1, 2, 3} are shown in Figs. 11,12 and 13. The classification accuracies were shown to be $\{\ 84.38\%, 82.81\%, 82.03\%\}$. This 4-class training set were based on 32 samples of the four classess given. The diagonal entries for the three matrices from kNN , $\{k = 1,2,3\}$ were {26, 26, 29, 27}, {22, 30, 27, 27}, & {24, 26, 26, 29}. The classification accuracies were computed based on the same formular in Equation 6.1. Out 32 samples from $C1$ class shown in Fig. 11 , 1 sample, 2 samples, and 3 samples were mispredicted for classes 2, 3, and 4 respectively. The same explaination applies to Figs. 12 and 13.

6.2 Receiver operating characteristics (ROC)

The receiver operating characteristic is a performance metric used to check the quality of classifiers(learning machine models). For each class of a classifier, threshold values across the interval [0,1] are applied to outputs. For each threshold, two values are calculated, the True Positive Ratio (the number of outputs greater or equal to the threshold, divided by the number of one targets), and the False Positive Ratio (the number of outputs greater

than the threshold, divided by the number of zero targets). Thus, the ROC (for any classifier) is the graphical plot of True Positive Rate (TPR) against False Positive Rate (FPR) or sensitivity against (1-specificity). TPR is the same thing as sensitivity and FPR + specivicity = 1. The ROCs for the classifier used in this work is shown in Fig. 14. In the ROCs figure shown, TPR of all classes is ploted against the FPR of all classes. The varying parameters along each ROC is TPR and FPR of all the number of pattern (instances) in each class. The the number of observations (instances) for each image of the cells varies. The TPR of all ROC curves generated by the kNN for the 4{8}-class training set all lie between 0.78 and 1. The average TPR for all the classes is 0.8325 while average FPR for all the classes is 0.0699. A perfect classifier should have (0, 1) for this ordered pair. The same pair for the kNN has value {(0.0699, 0.8325)}. This indicates a good performance for the classification model in this study. The metrics for the other two classifier are shown in Fig. 15.

Fig. 7. Image classification tool for cell cultures identification. The test cell was PC3. The classified cell was correctly identified as PC3

	Class 1	Class 2	Class 3	Class 4	Class 5	Class 6	Class 7	Class 8
Class 1	14	0	0	0	1	0	1	0
Class 2	0	11	0	0	0	4	0	1
Class 3	1	0	13	0	0	0	2	0
Class 4	0	0	0	15	0	0	0	1
Class 5	0	0	0	0	15	0	1	0
Class 6	0	3	0	1	0	11	0	1
Class 7	0	0	0	0	0	0	16	0
Class 8	0	2	0	1	1	0	0	12

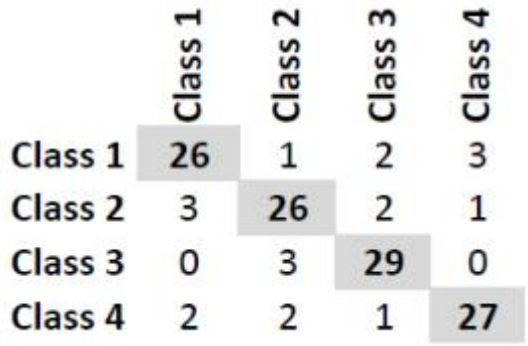
Fig. 8. Confusion Matrix for 8-class Training set: Run1 (kNN, k = 1, Accuracy = 83.59%. Class 1 = HeLa (BF), Class 2 = HeLa(DAPI), Class 3 = PC3 (BF), Class 4 = PC3 (DAPI), Class 5 = BHK (BF), Class 6 = BHK (DAPI), Class = Vero (BF), Class 8 = Vero (DAPI)

	Class 1	Class 2	Class 3	Class 4	Class 5	Class 6	Class 7	Class 8
Class 1	**14**	0	1	0	1	0	0	0
Class 2	0	**11**	0	0	0	2	0	3
Class 3	1	0	**12**	0	0	0	3	0
Class 4	0	0	0	**15**	0	1	0	0
Class 5	0	0	0	0	**15**	0	1	0
Class 6	0	4	0	1	0	**10**	0	1
Class 7	1	0	0	0	1	0	**14**	0
Class 8	0	0	0	2	0	0	0	**14**

Fig. 9. Confusion Matrix for 8-class Training set: Run2 (kNN, k = 2, Accuracy = 82.03%. . Class 1 = HeLa (BF), Class 2 = HeLa(DAPI), Class 3 = PC3 (BF), Class 4 = PC3 (DAPI), Class 5 = BHK (BF), Class 6 = BHK (DAPI), Class = Vero (BF), Class 8 = Vero (DAPI))

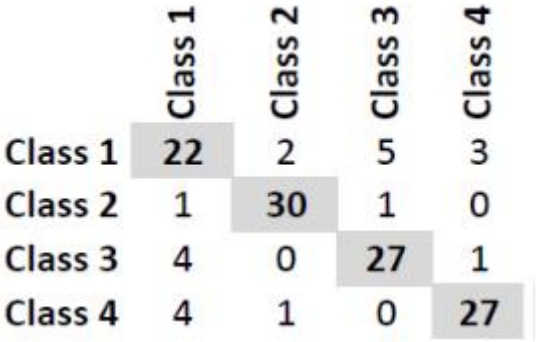
	Class 1	Class 2	Class 3	Class 4	Class 5	Class 6	Class 7	Class 8
Class 1	**15**	0	0	0	1	0	0	0
Class 2	0	**12**	0	0	0	1	0	3
Class 3	2	0	**14**	0	0	0	0	0
Class 4	0	0	0	**15**	0	0	0	1
Class 5	1	0	1	0	**14**	0	0	0
Class 6	0	5	0	1	0	**9**	0	1
Class 7	1	0	0	0	1	0	**14**	0
Class 8	0	3	0	1	0	1	0	**11**

Fig. 10. Confusion Matrix for 8-class Training set: Run3 (kNN, k = 3, Accuracy = 81.25%. Class 1 = HeLa (BF), Class 2 = HeLa(DAPI), Class 3 = PC3 (BF), Class 4 = PC3 (DAPI), Class 5 = BHK (BF), Class 6 = BHK (DAPI), Class = Vero (BF), Class 8 = Vero (DAPI))



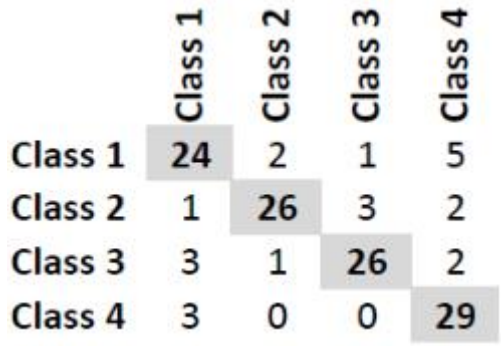
	Class 1	Class 2	Class 3	Class 4
Class 1	26	1	2	3
Class 2	3	26	2	1
Class 3	0	3	29	0
Class 4	2	2	1	27

Fig. 11. Confusion Matrix for 4-class Training set: Run1 (kNN, k = 1, Accuracy = 84.38%. Class 1 = HeLa, Class 2 = PC3, Class 3 = BHK, Class 4 = Vero)



	Class 1	Class 2	Class 3	Class 4
Class 1	22	2	5	3
Class 2	1	30	1	0
Class 3	4	0	27	1
Class 4	4	1	0	27

Fig. 12. Confusion Matrix for 4-class Training set: Run2 (kNN, k = 2, Accuracy = 82.81% Class 1 = HeLa, Class 2 = PC3, Class 3 = BHK, Class 4 = Vero)



	Class 1	Class 2	Class 3	Class 4
Class 1	24	2	1	5
Class 2	1	26	3	2
Class 3	3	1	26	2
Class 4	3	0	0	29

Fig. 13. Confusion Matrix for 4-class Training set: Run3 (kNN, k = 3, Accuracy = 82.03% Class 1 = HeLa, Class 2 = PC3, Class 3 = BHK, Class 4 = Vero)

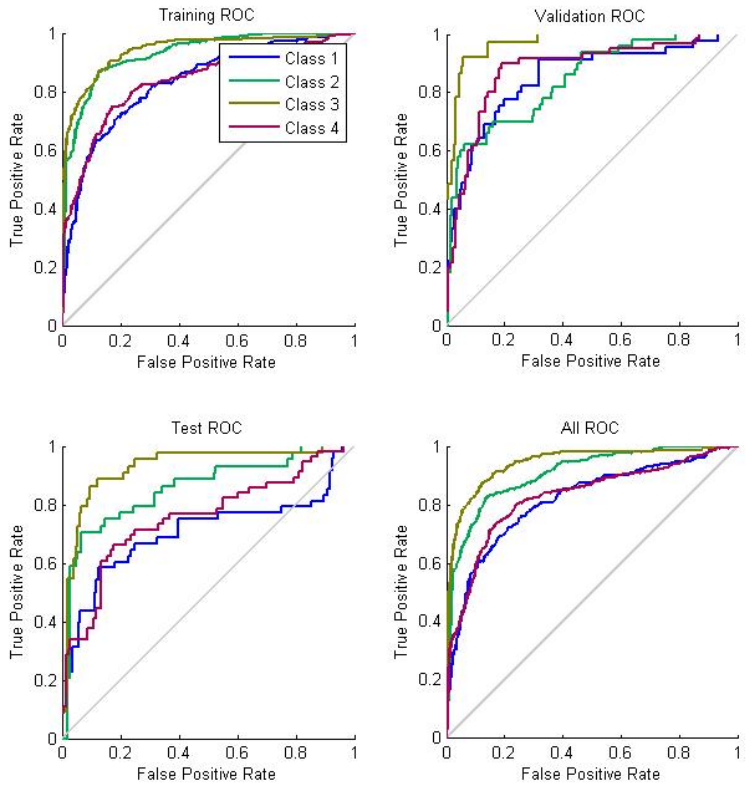


Fig. 14. Receiver Operating Chracteristics for our classification model

Metric	kNN	CNN	MLP
Accuracy(%)	84.38	87.25	86
TPR	0.8438	0.8724	0.8659

Fig. 15. Performance metric of the classifiers

7 Conclusion and Future Directions

The main objective of this work was to develop an adaptive computer-based vision system to accurately identify four cell types that are commonly employed in biomedical and virology laboratories. Extracted features were circularity, LM and ZM. Some of images in the database were subjected to translation, scaling and rotation. Our system was able to correctly classify the cell types with up to accuracies of 84.38%, 86%, 87.25% for kNN, MLP and CNN respectively. Our approach could be extended to find cells undergoing mitosis, analyse images of chromosome arrangement into karyograms, and classify chromosomal features in images, including detection of genomic defects. Further, this work may be extended to classify the composition of cell mixtures within a single microscopy image. Finally, image-based systems biology involves the combination of systematic quantitative image data collection with spatiotemporal systems modelling ([52], [53]). These approaches may enable the classification of cell behaviors that span spatial scales from molecular to cellular and to tissue level. Inclusion of several segmentation techniques and filter-based image analysis would worth considering in the future works.

Acknowledgements

We are grateful for support from the US National Institute of Health and the Graduate School of the University of Wisconsin, Madison, WI, USA (Grant Nos, AI091646 and AI104317 respectively).

Competing Interests

Authors have declared that no competing interests exist.

References

[1] Warrick J, Timm A, Swick A, Yin J. Tools for single-cell kinetic analysis of virus-host interactions. PLoS One. 2016;11(1):e0145081.

[2] Ji N, Shroff H, Zhong H, Betzig E. Advances in the speed and resolution of light microscopy. Curr Opin Neurobiol. 2008;18(6):605-616.

[3] Weber M. Mickoleit M, Huisken J. Light sheet microscopy. Methods. Cell Biol. 2014;123:193-215.

[4] Stegmaier J, Amat F, William LC, McDole K, Wan Y, Teodoro G, Mikut R, Keller JP. Real-time three-dimensional cell segmentation in large-scale microscopy data of developing embryos. Developmental Cell Technology; 2016.

[5] Boland MV, Borland RV, Murphy RF. A neural network classifier capable of recognizing the patterns of all major subcellular structures in flourescence microscope images of HeLa cells. Bioinformatics. 2001;17(12):1213-1223.

[6] Danuser G. Computer vision in cell biology. Cell. 2011;147:973-978.

[7] Eliceri KW, Berthold MR, Golberg IG, Ibanez L, et al. Biological imaging software tools. Nat Methods. 2012;9:697-710.

[8] Javidi B, Rawat S, Komatsu S, Markan A. Cell Identification using single beam lensless imaging with pseudo-random phase encoding. Opt Lett. 2016;41(15):3663-3666.

[9] Khashman A. Blood cell identification using a simple neural network. Int J Neural Syst. 2016;18(5):453-458.

[10] Shuai H, Hu Y, Yu Q, Gylfe E, Tengholm A. Fluorescent protein vectors for pancreatic islet cell identification in live-cell imaging. Signalling and cell biology; 2016.
DOI: 10.1007/s00424-016-1864-z

[11] Zhou ZH, Jiang Y, Yang YB, Chen SF. Lung cancer cell identification based on artificial neural network ensembles. Artificial Intelligence in Medicine. 2002;24(1):25-36.

[12] Chong CW, Raveendran P, Mukundan R. Translation and scale invariants of legendre moments. Pattern Recognition. 2004;37(1):119-129.

[13] Nixon MS, Aguado AS. Feature extraction & image processing for computer vision. Elseviet Ltd, the Boulevard, Langford Lane, Kindlington, Oxford, OX5 1GB, UK; 2012.

[14] Held M, Schmitz MH, Fischer B, Olma MH, Matthias P, Jan E, Daniel WG. CellCognition: Time-resolved phenotype annotation in high-throughput live cell imaging. Nature Methods. 2010;7(9):747-754.

[15] Pau Gregoire, Florian Fuchs, Oleg Sklyar, Michael Boutros, Wolfgang Huber. EB Image - An R package for image processing with applications to cellular phenotypes. BioInformatics. 2010;26(7):979-981.

[16] Flint SJ, Flint VR, Racantiello LW, Enquist LW, Skaika AM. Molecular biology: Principles of virology. Volume I, 4th Edition; 2015.

[17] Meng W, Zhou X, King WR, Stephen TW. Context based mixture model for cell phase identification in automated flourescence microscopy. BMC Bioinformatics. 2007; 8(32).
DOI: 10.1186/1471-1205-8-32

[18] Skloot R. The immortal life of Henrietta Lacks. Published by Pan Macmillan; 2010.
ISBN: 978-0-330-533447.

[19] Rahbari R, Sheahan T, Modes V, Collier P, Macfariane C, Badge RM, Sheahan M. A novel L1 retrotransposon marker for HeLa cell line identification. BioTechniques. 2009;46(4):277-284.

[20] Fang T. ATCC Biological controls for the detection and analysis of cancer. ATCC presentation; 2014.

[21] Yasumura Y, Kawaikita M. The research for the SV40 by means of tissue culture technique. Nippon Rinsho. 1963;21(6):1201-1219.

[22] Osada N, Kohara A, Yamaji T, Hirayama N, Kasai F, Sekizuka T, Kuroda M, Hanada K. The genome landscape of the African green monkey kidney-derived Vero cell line. DNA Research. 2014;21:673-83.

[23] Lam V, Duca KA, Yin J. Arrested spread of vesicular stomatitis virus infections in vitro depends on interferon-mediated antiviral activity. Biotechnol Bioeng. 2005;90(7):793-804.

[24] Alimirah FC, Basrawala, Xin H, Choubey D. PC-3 cell line expresses androgen receptor: implications for the androgen receptor functions and regulation. FEBS Lett. 2016;580(9):2294-300.

[25] Ghosh A, Xinning W, Klien E, Warren DWH. Novel role of prostrate-specific membrane antigen in suppressing prostrate cancer invasiveness. Cancer Research. 2005; 65(3):727-31.

[26] Kaighn ME, Narayan YO, Lechner JF, Jones LW. Establishment and charaterization of a human prostatic carcinoma cell line (PC-3). Invest Urol. 1979;17(1):16-23.

[27] Pulukuri SM, Gondi CS, Lakka SS, Ama J, Norman E, Meeta G, Jasti SR. RNA interference-directed knockdown of urokinase plasminogen activator and urokinase plasminogen activator receptor inhibits prostrate cancer cell invasion, survival, and tumorigenicity in vivo. J. Biol. Chem. 2005;280(43):36529-40.

[28] Tai S, Sun S, Squires JM, Zhang JH, OH, WK, Liang WK, CZ, Huang J. PC3 is a cell line characteristic of prostate small cell carcinoma. The prostate. 2011;71(15):1668-1679.
Available: http://doi.org/10.002/pros.21383

[29] Ahmed MM, Ahmed SD, Cramer SD, Lyles DS. Sensitivity of prostate tumors to wild type and M protein mutant vesicular stomatitis viruses. Virology. 2004;330(1):34-49.

[30] Swick A, Baltes A, Yin J. Visualizing infection spread: Dual-color fluorescent reporting of virus-host interactions. Biotechnol Bioeng. 2014;111(6):1200-1209.

[31] Zhang J, Tan T. Brief review of invariant texture analysis methods. Pattern recognition. 2002;35(3):735-47.

[32] Flusser J. On the independence of rotation moment invariants. Pattern Recognition. 2000;33:1405-1410.

[33] Flusser J, Suk, Zitova B. Moments and moment invariants in pattern recognition. A John Wiley and Sons, Ltd, Publication. 2009;1-303.

[34] Babatunde O, Armstrong L, Leng J, Diepeveen D. Zernike moments and genetic algorithm: Tutorial and application. British Journal of Mathematics & Computer Science. 2014;4(15):2217-2236.

[35] Liao SX, Pawlak M. On image analysis by moments. IEEE Transactions on Pattern Analysis and Machine Intelligence. 1996;18(3):254-66.

[36] Hosny KM. Exact legendre moment computation for gray level images. Pattern Recognition. 2007;40(12):3597-3605.

[37] Tech CH, Chin RT. On image analysis by the methods of moments. Pattern Analysis and Machine Intelligence, IEEE Transactions on. 1998;10(4):496-513.

[38] Richard JP, Prokop J, Anthony PR. A survey of moment-based techniques for unoccluded object representation and recognition, CvGiP; Graphical Models and Image Processing. 1992;54(5):438-460.

[39] Khotanzad A, Hong YH. Invariant image recognition by Zernike moments. In IEEE Transactions on Pattern Analysis and Machine Intelligence. 1990;12(5):489-497.
DOI: 10.1109/34.55109

[40] Zink D, Sadoni N, Stekzer E. Visualizing chromation and chromosomes in living cells. Methods. 2003;29(1):42-50. PMID 125430070.
DOI: 10.1016/S1046-2023(02)00289-X

[41] Maksymilian P. Advanced light microscopy: Principles and basic properties. Elsevier. 1988;1.

[42] Maksymilian P. Advanced light microscopy: Principles and basic properties. Elsevier. 1998;2.

[43] Babatunde O. A neuro-genetic hybrid approach to automatic identification of plant leaves. PhD Thesis in The School of Computer and Security Science, Edith Cowan University, Perth, WA, Australia. 2015;1-300.

[44] Cover T, Hart P. Nearest neighbor pattern classification. Information Theory, IEEE Transactions on. 1967;13(1):21-27.

[45] Wang K, Chen T, Lau R. Bagging for robust non-linear multivariate calibration of spectroscopy. Chemometrics and Intelligent Laboratory Systems. 2011;105(1):1-6.

[46] Bishop CM. Neural networks for pattern recognition. Oxford University Press; 1995.

[47] Demuth H, Beale M, Hagan M. Neural network toolbox users' guide. The MathWorks.Inc., Natick, MA, USA; 2013.

[48] Russel S, Norvig P. Artificial intelligence: A modern approach (M. J. Horton, Ed.). Prentice Hall Series in Artificial Intelligence; 2003.

[49] Havael M, Davy A, Warde-Farley D, Biard A, Courville A, Bengio Y, Larochelle H. Brain tumor segmentation with deep neural networks. Medical image analysis. 2017;35:18-31.

[50] Zeiler MD, Fergus R. Visualizing and understanding convolutional networks. In: Computer Vision-ECCV 2014. Springer. 2014;818-833.

[51] Mathworks. Statistics toolbox: Users guide for version r2013a. The MathWorks, Inc. 3 Apple Hill Drive Natick, MA 01760-2098; 2013.

[52] Haseline EL, Lam V, Yin J, Rawlings JB. Image-guided modeling of virus growth and spread. B Math Biol. 2008;70(6):1730-1748.

[53] Lam V, Boehme KW, Compton T, Yin J. Spatial patterns of protein expression in focal infections of human cytomegalovirus. Biotechnol Bioeng. 2006; 93(6):1029-1039.

Peer-review history:
The peer review history for this paper can be accessed here (Please copy paste the total link in your browser address bar)
http://sciencedomain.org/review-history/19742