Microscopy cell images classification. An overview of supervised Machine & Deep Learning methods

Study carried out by the Data Science Practice

Special thanks to Ali AHMAD

canopee

empowering *ecosystem*

www.canopee-group.com

Summary

Introduction

Microscopy plays an indispensable role in biomedical research. With the development of optics and informatics, advanced microscopy technologies have opened a new perspective for biomedical researchers. Potentially, the power of these technological developments in microscopy is increased tenfold when combined with the diversity of fluorescence microscopy modalities and artificial intelligence through machine and deep learning methods. An important biological application of microscopy and artificial intelligence is the classification of cells to recognize their phenotypes based on their individual spatial characteristics such as shape, texture, size, etc.

In this note, we aim to present an overview of supervised classical machine learning and deep learning approaches used for 2D cell image classification. First, we show the classification workflow in the traditional machine learning approach as well as in the deep learning approach. Then, we describe the most commonly used methods. Finally, we compare these methods for a real use case of cancer cell image classification.

1 Classical machine learning VS deep learning : What is the key differences ?

Classical machine learning and deep learning are subfields of artificial intelligence. Both machine learning and deep learning methods tend to automate repetitive tasks, with the least amount of error , by using a set of learning algorithms (supervised or unsupervised learning processes) to analyze data, interpret it, learn from it, and make the best possible decisions based on those learnings [1].

Classical machine learning algorithms use statistical strategies and fundamental principles of computer science to make machines learn from structured data (features). The traditional machine learning workflow includes human intervention and uses traditional handcrafted feature definition to solve a problem. However, deep learning structures algorithms in layers to create an "artificial neural network" that can learn and make intelligent decisions on its own directly from large volumes of unstructured data such as images, audio, or video. These deep learning networks attempt to learn high-level features from data incrementally and eliminate the need for domain expertise and basic feature extraction (see Figure 1). For instance, if a traditional machine learning algorithm returns an inaccurate prediction, a machine learning expert must step in and adjust features and other variables to achieve accuracy. In contrast, deep learning algorithms can determine whether or not the prediction is accurate and adjust its weights automatically through the network.

Another key difference between deep learning and traditional machine learning is its performance as the scale of the data increases. While machine learning algorithms work with small to medium-sized data, larger data is required for deep learning algorithms. Thus, when the data is small, deep learning algorithms do not perform as well, with a risk of overfitting.

2 Classical machine learning methods for image classification

In this section, we describe the steps of the workflow of traditional machine learning represented in the Figure 1. We describe the feature extraction methods commonly used for 2D image classification and present the classification algorithms that are eventually used to classify images from the extracted features.

2.1 Feature Extraction in image processing

In the field of computer vision and image processing, feature extraction is the process of transforming the raw pixel values of an image into more meaningful and useful information that can be used directly in machine learning. It yields better results than applying machine learning directly to the raw input image, which is usually large and contains complex and redundant information. Thus, feature extraction step is considered part of the dimensionality reduction process [2].

There are several types of features that can be processed from an input image, such as shape, color, texture, pointillist features, etc [3, 4]. These features are selected based on the application and content of the input image to be classified. In this note, we will focus only on describing the commonly used textural feature methods for microscopy image classification.

2.2 Textural features extraction methods

There is a wide range of textural features extraction methods [5] and there is no proof of optimality for any tool. The most popular methods are detailed in the following.

Gray Level Co-ocurrence Matrix (GLCM) Gray level co-occurrence matrix (GLCM) is a classical statistical approach that can well describe second-order statistics of a texture image. GLCM was firstly introduced by [6] and is essentially a two-dimensional histogram in which the $(i, j)th$ element is the frequency of pixel intensity i co-occurring with pixel intensity j. A co-occurrence matrix P has a dimension of $n \times n$, where n represent the gray level in an image and it is specified by the relative frequencies $C(i, j, d, \theta)$ in which two pixels, separated by a distance d, occurs in a direction specified by the angle θ (usually θ : 0, 45, 90 and 135 degrees), one with gray level i and the other with gray level j (see Figure 2). To characterize the textures in an image, a set of 14 Haralick coefficients summarizing the co-occurence matrix P is then computed. They are detailed in the following :

1. Sum average :

$$\sum_{i=1}^{2n} iP_{x+y}(i) \tag{1}$$

2. Sum variance :

$$\sum_{i=1}^{2n} (i - \mu_{x+y})^2 P_{x+y}(i) \tag{2}$$

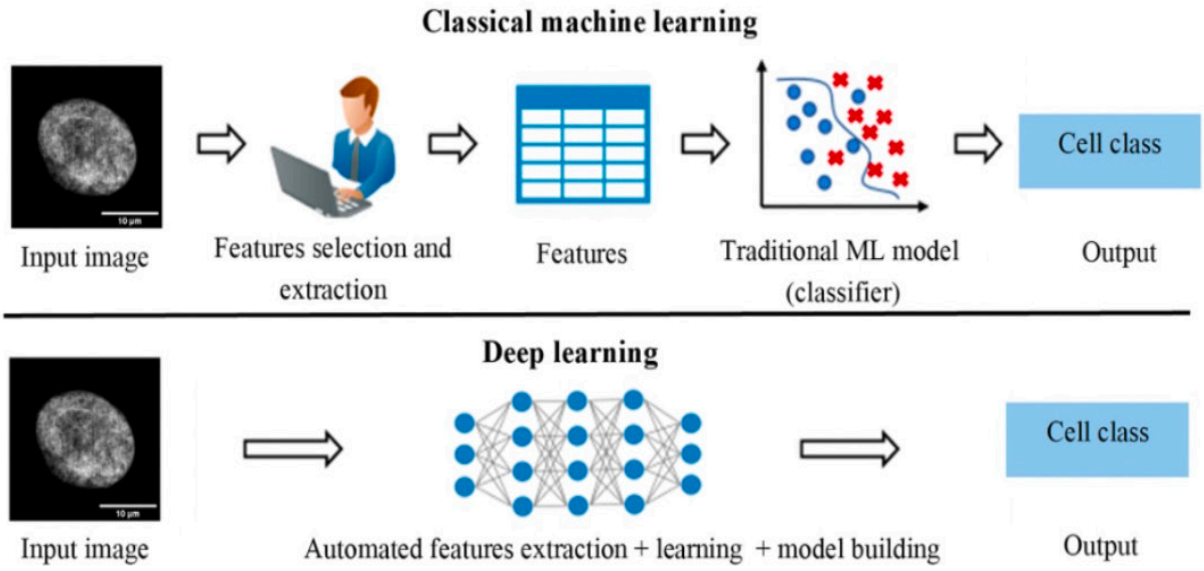

FIGURE 1 – Classical machine learning VS Deep learning workflow.

3. Contrast :

$$\sum_{i=1}^{n}\sum_{j=1}^{n}(i-j)^2 P(i,j) \tag{3}$$

4. Correlation :

$$\frac{\sum_{i=1}^{n}\sum_{j=1}^{n} ijP(i,j) - \mu_x\mu_y)}{\sigma_x\sigma_y} \tag{4}$$

5. Entropy :

$$-\sum_{i=1}^{n}\sum_{j=1}^{n} P(i,j)log(P(i,j)) \tag{5}$$

6. Energy :

$$\sum_{i=1}^{n}\sum_{j=1}^{n} P(i,j)^2 \tag{6}$$

7. Dissimilarity :

$$\sum_{i=1}^{n}\sum_{j=1}^{n} |i-j|P(i,j) \tag{7}$$

8. Difference entropy :

$$-\sum_{i=0}^{n-1} P_{x-y}(i)log(P_{x-y}(i)) \tag{8}$$

9. Difference variance :

$$\sum_{i=0}^{n-1}(i-\mu x-y)^2 P_{x-y}(i) \tag{9}$$

10. Information measure of correlation 1 :

$$\frac{HXY - HXY1}{max(HX, HY)} \tag{10}$$

11. Information measure of correlation 2 :

$$\sqrt{1-exp[-2(HXY2-HXY)]} \tag{11}$$

12. Inverse difference :

$$\sum_{i=1}^{n}\sum_{j=1}^{n}\frac{P(i,j)}{1+|i-j|} \tag{12}$$

13. Sum entropy :

$$-\sum_{i=2}^{2n} P_{x+y}(i)log(P_{x+y}(i)) \tag{13}$$

14. Sum variance :

$$-\sum_{i=2}^{2n}(i-\mu_{x+y})^2 P_{x+y}(i) \tag{14}$$

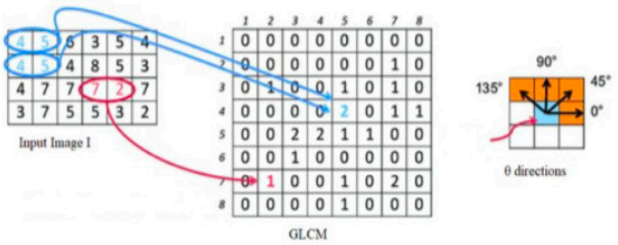

FIGURE 2 – Principle of the computation of the gray level co-occurrence matrix (GLCM).

In order to calculate the above features, it is necessary to define the following statistics :

$$HX = -\sum_{i=1}^{n} P_x(i)log(P_x(i))$$

$$HX = -\sum_{i=1}^{n} P_y(i)log(P_y(i))$$

$$HXY = -\sum_{i=1}^{n}\sum_{j=1}^{n} P(i,j)log(P(i,j))$$

$$HXY1 = -\sum_{i=1}^{n}\sum_{j=1}^{n} P_x(i,j)log(P_x(i)P_y(j))$$

$$HXY2 = -\sum_{i=1}^{n}\sum_{j=1}^{n} P_x(i)P_y(j)log(P_x(i)P_y(j))$$

$$P_x(i) = \sum_{j=1}^{n} P(i,j)$$

$$P_y(j) = \sum_{i=1}^{n} P(i,j)$$

$$P_{x+y}(i) = \sum_{i+j=q} P(i,j), q = 2,3,\dots 2n$$

$$P_{|x-y|}(i) = \sum_{i+j=q} P(i,j), q = 0,1,2,\dots n-1$$

$$\mu_x = \sum_{i=1}^{n}\sum_{j=1}^{n} iP(i,j)$$

$$\mu_y = \sum_{i=1}^{n}\sum_{j=1}^{n} jP(i,j)$$

Local Binary Patterns (LBP) Local binary patterns are also among the most used texture descriptors in classification tasks [7]. For each central pixel position coordinate (x, y) of the image, the local binary pattern (LBP) indicates a sequential set of the binary comparison of its value with the P neighbors in a circle of radius R around the central pixel. The LBP assigns to each neighbor the value 0 or 1 if its value is smaller or greater than the pixel placed at the center, respectively. The resulting decimal value of the generated binary number replaces the central pixel value and it can be expressed as follows,

$$LBP(x,y) = \sum_{n=0}^{P-1} 2^n b\left(i_n - i_{x,y}\right), \tag{15}$$

where $i_{x,y}$ is the gray value of the central pixel and i_n denotes the n^{th} neighboring one. Besides, the function $b(z)$ is defined as follows,

$$b(z) = \begin{cases} 1, & \text{if } z \geq 0 \\ 0, & \text{if } z < 0\,. \end{cases} \tag{16}$$

The frequency of occurrences of each decimal code is then calculated over each region and used as a texture descriptor (see Figure 3).

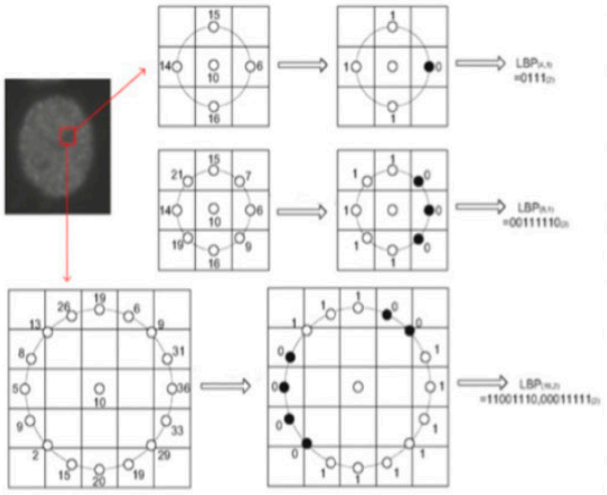

FIGURE 3 – Principle of extraction of LBP textural features from a 2D image with neighborhood and radius pairs : (P = 4, R = 1), (P = 8, R = 1) and (P = 16, R = 2) and the resulting decimal codes.

Scattering transform (SCATNET) A scattering transform defines a signal representation which is invariant to translations and potentially to other groups of transformations such as rotations or scaling. It is also stable to deformations and is thus well adapted to image and audio signal classification [8]. A scattering transform is implemented with a convolutional network architecture, iterating over wavelet decompositions and complex modulus. Figure 4 shows a schematic view of a scatter transform network working as a feature extractor. The scatter vectors Z_m at the output of the first three layers m = 1, 2, 3 for an input image f are defined by,

$$\begin{aligned} Z_1 f &= \{|f| * \phi\} \\ Z_2 f &= \{\cdots, |f * \psi_{j,\theta}| * \phi, \cdots\} \\ Z_3 f &= \{\cdots, ||f * \psi_{j,\theta}|\psi_{k,\varphi}| * \phi, \cdots\}, \end{aligned} \tag{17}$$

where the symbol $*$ denotes the spatial convolution, $|\cdot|$ stands for the L_1 norm, ϕ is an averaging operator, $\psi_{j,\theta}$ is a wavelet dilated by 2^j and rotated by θ. The range of scales $j = \{1, \cdots, J\}$ and the number of orientations $\theta = \{0, \pi/L, \cdots, \pi(L-1)/L\}$ are fixed by integers J and L. The number of layers is between $m = 1$ to $m = M$. A visualization of the Gabor filter bank used to extract SCATNET features and of a cell image at the output of the scattering transform array with 3 layers (m) for 4 scales (J) and 8 orientations (L) is shown in Figure 5. A Matlab toolbox for extracting features from images by scattering transform is available here : SCATNET.

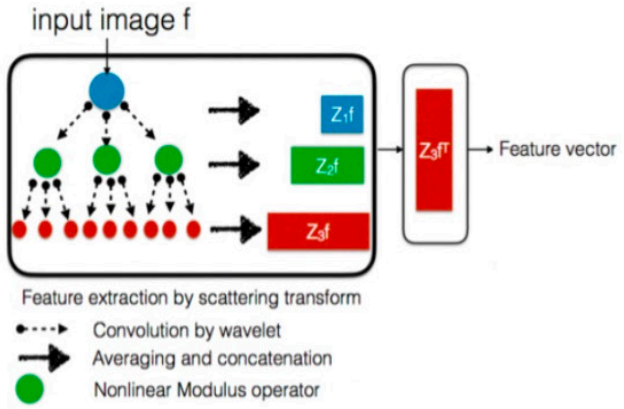

FIGURE 4 – Schematic layout of the images feature extraction based on the scattering transform with three layers. The feature vector at the output is the scatter vector $Z_m f$ of the last layer of Equation 17 after transposition.

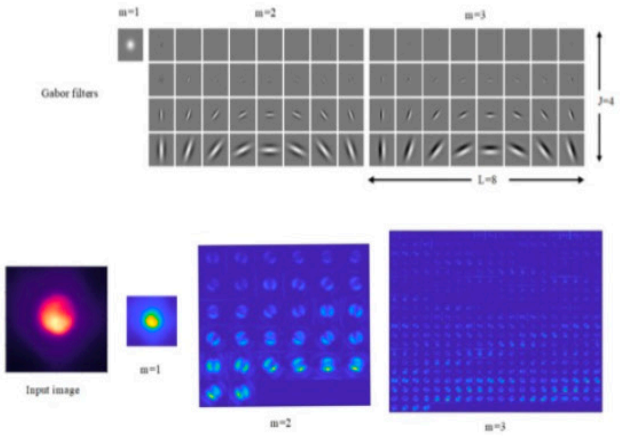

FIGURE 5 – Filter bank and images at the output of scattering transform network for each layer m of the scatter transform.

2.3 Classifiers

In the classical image classification pipeline, classification tasks are approached based on a classifier that is trained on the extracted features to automatically classify the images into two or more sets of classes. There are many classification algorithms available in the literature, such as decision tree, random forest, logistic regression classifier, Bayes classifier, support vector machines (SVM), K-nearest neighbors, etc. [9] but there is no proof of optimality for any classifier. The classification task generally depends on the application and the nature of the available dataset.

In this section, we will focus on the widely used classifier for image classification which is the Support Vector Machine (SVM) [10]. Initially, the SVM was designed as a classification algorithm that can only handle linear classification problems. SVMs are based on the idea of finding an hyperplane that best divides a dataset into two classes, as shown in Figure 6.A. it aims to separate the data using a boundary as simple as possible, so that the distance between the different groups of data and the boundary that separates them is maximum. This distance is also called the margin and the support vectors being the data closest to the frontier. The technique of margin maximization makes it possible to guarantee a better robustness to noise and thus a more generalizable model. The notion of boundary assumes that the data are linearly separable, which is rarely the case. To overcome this, researchers improved the SVMs to solve the non-linear classification problems by relying on the use of kernels. These mathematical functions allow to separate the data by projecting them into a higher dimensional vector space (see Figure 6.B).

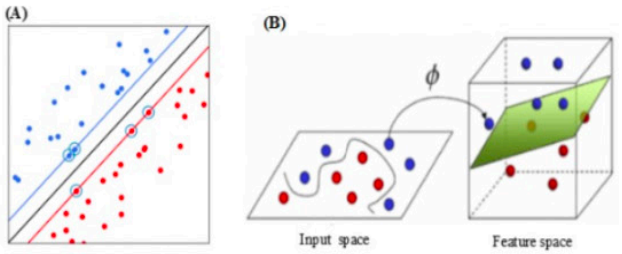

FIGURE 6 – Principle of Support Vector Machines (SVMs). (A) In this two-dimensional space, the "boundary" is the black line, the "support vectors" are the surrounded points (closest to the boundary) and the "margin" is the distance between the boundary and the blue and red lines. (B) SVMs allow to project the data into a higher dimensional space via a kernel function to separate them linearly.

3 Classification based deep learning method

Deep learning based image classification algorithms mostly relys on the Convolutional neural networks (CNN). They are algorithms, which are particularly useful for image analysis, have more complex network structure and more powerful feature learning and feature expression abilities than traditional machine learning methods [11]. They are the fundamental and basic building blocks for the most existing deep learning architectures used for image recognition such as the DenseNet [12], VGG16 [13], ResNet [14], etc. The structure of a CNN consists of a succession of layers : an input layer (the

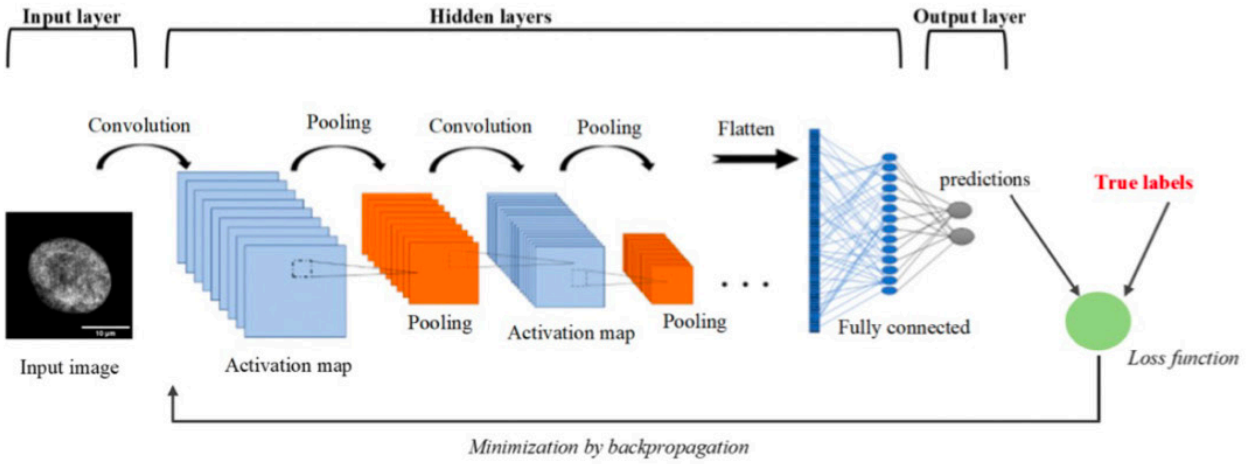

FIGURE 7 – Basic architecture of a convolutional neural network (CNN).

image at the input of the network), an output layer with an output activation function (the decision of the network) and a hidden layer composed of many convolutional layers, correction layers (activation functions), pooling layers and fully connected or dense layers (see Figure 7). The description of the components of a CNN architecture is as follows :

- *The convolutional layer :* is the key component of convolutional neural networks. It is used to extract a set of features in the images received at the input of the CNN by a filtering operation (with n filters). At the output of this layer, activation maps are generated indicating where are located the pixels with the most discriminating descriptors. Contrary to traditional methods, the filter weights are not pre-defined according to a particular formalism but learned by the network during the training phase. These weights are randomly initialized and then optimized by backpropagation of the gradient.
- *The activation function :* is a key part of CNN design. It is a non-linear function applied to the activation maps at the output of the convolution layers. By making the data non-linear, it facilitates the extraction of complex features. The modern default mostly used activation function for convolutional layers is the Rectified Linear Activation (ReLU) function as it is less susceptible to vanishing gradients that prevent deep models from being trained.
- *The pooling layer :* it consists in reducing the image size while preserving their important characteristics. It reduces the number of parameters and computations in the network. This improves efficiency and avoids overfitting. Moreover, this pooling layer allows the extraction of contexts at different image scales.
- *The fully connected layer (FC) :* it is used as a decision layer in the CNN architecture. It consist of a set of neurons that input vector containing the pixels of the flattened, corrected and reduced images by pooling. The weights of this layer (FC) are adjusted in the same way as the weights of the filters of the convolution layer : during the training phase, by backpropagation of the gradient.
- *The output activation function :* the choice of activation function at the output layer will define the type of predictions the model can make. For exemple, in regression problems, the linear output activation function is typically used to directly return the weighted sum of the input. However, in classification problems, the softmax function is widely used to produce a vector of values that sum to 1.0 and can be interpreted as class probabilities.
- *The loss function :* it specifies how the training of the network penalizes the difference between the prediction and the true values. Various loss functions adapted to different tasks can be used. For a classification problem, for instance, we cite the catergorial cross-entropy (CCE, Eq. 18) for multi-class image classification problems, and the binary cross-entropy (BCE, Eq. 19), which is a specific case of the CCE, for binary classification tasks (only two classes : [0,1]).
  The CCE is defined as follow,

$$CCE = -\sum_{c=1}^{J} y_{i,c}\ log(p_{i,c}), \tag{18}$$

  with, J denotes the number of classes, $y_{i,c}$ is a binary indicator (0 or 1) that indicates whether the class c for the sample i is correct and $p_{i,c}$ denotes the predicted probability for that sample. While the BCE could be defined as the following,

$$BCE = -[y_i\ log(p_i) + (1 - y_i)\ log(1 - p_i)]. \tag{19}$$

4 Use Case : classification of cancer cells

We compared the performance of each previously described classification method on the basis of a use case of high-throughput real time cancer cell screening. The dataset used,

the configuration of the classification methods, and the results are described below.

4.1 Image acquisition and dataset

The cells used in this use case are hTERT-immortalized human mammary epithelial cells (IMEC WT), xenograft-derived primary tumor cells (XD), and lung metastasis-derived cells (MD). All cell lines are transduced with PGK-H2B mCherry lentiviral vector (central emission wavelength λ_{em} = 610 *nm*), as result cells express the mCherry H2B recombinant protein in the nucleus. Furthermore, the XD and MD cell lines were obtained by injecting orthotopically IMEC cells overexpressing MYC carrying also the PIK3CA H1047R mutation in NOD/SCID mice as described in [15].

Multi-class cell images were acquired with a microfluidic light-sheet fluorescence microscopy system [16] with a cell flow speed of 140 nl/min corresponding to 5 $cells/s$. The total amount of the used dataset in this study is 1380, 2236 and 1890 images for WT, XD and MD cell types respectively. An illustration of the 2D cell images is shown in the Figure 8.

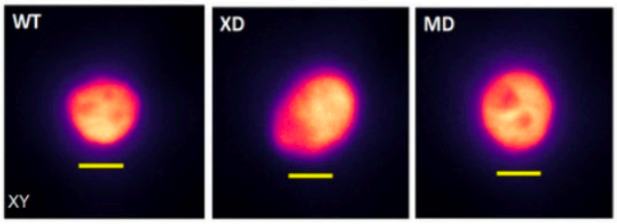

FIGURE 8 – An example of 2D images of the three cell types. Yellow scale bar = 5 μm.

4.2 Configuration of classification methods

In order to compare the classification of the three cell classes, we propose to perform the classification on textural feature spaces followed by an SVM classifier with a cubic kernel (chosen as it gives the highest classification performance) [10]. We also add a deep learning based convolutional neural network (CNN) for 2D image classification. We empirically optimize the hyperparameters of the used methods as well as the architecture of the CNN and they are implemented as follows :

- *GLCM :* gray level co-occurance matrix were computed from each cell image with a neighbhor distance d = 16 pixels. 14 haralick coeficients were then computed for each cell image. These features were used to train and test the SVM classifier.
- *LBP :* for the present use case, the local binary pattern features were extracted with the neighborhood size P, optimized empirically and found to be optimal at P = 16, while the radii was set to R = 2.
- *SCATNET :* we used the Gabor filter as the mother wavelet with the scattering transform parameters that were optimized in an empirical way with 3 layers for 4 scales and 8 orientations of the filters (see Figure 5).
- *Deep learning CNN :* we used here a VGG16 [13] for the classification of the 2D images trained with a Tesla V100-DGXS-32GB GPU. Briefly, a VGG16 architecture consist of 13 convolution layers divided to 5 convolution blocks and 3 fully connected layers with (4096,4096,*nc*) neurons respectively, with *nc* is the number of classes equal to 3. It has a Max pooling layer of size 2 × 2 for each convolution block. It uses the softmax function as the output layer and ReLu activation function is applied to all hidden layers. We trained the VGG16 model with the following optimized hyper parameters : filter size=3 × 3, filters number for the convolution blocks respectively= (64, 128, 256, 512, 512), batch size= 32, number of training epochs= 100, learning rate= 0.0001.
  In order to provide sufficient amounts of data, a data augmentation step was used on the training sets. The augmentation operation contained only geometric transformations such as horizontal and vertical flipping and a random rotational transformation between 0 and 359° (see Figure 9). In addition, a regularisation step based on early stopping and best model weights saving was added during the training to avoid the model over-fitting. We used the categorical cross entropy (CCE, see Eq. 18) as a loss function during model training. A weight were also automatically included for each cell class during loss function computation to deal with the class imbalance.

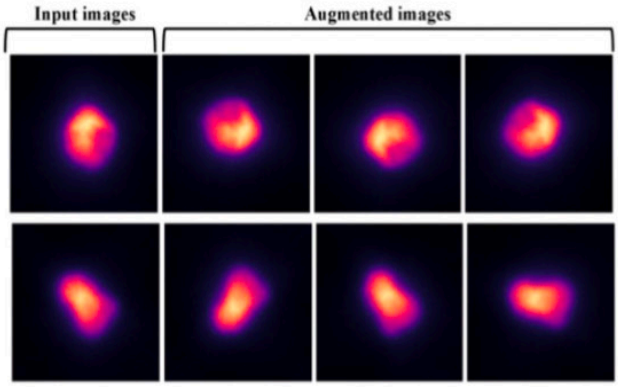

FIGURE 9 – Examples of image augmentation based on geometrical transformations.

The feature extractions and classification pipeline has been applied for each cell image. The number of extracted features of each described method is shown in the Table 1. In order to deal with the class imbalance, we used the stratified 10-folds cross-validation method to quantify the classification accuracy. The final accuracy performance was computed as the average of the measured 10-folds accuracies for each method.

TABLE 1 – Images classification performances (% of Accuracy), number of features and computational time (in seconds computed from test images) for textural feature spaces and CNN methods. Gray-colored table cells represents the highest accuracy values and the green colored table cell is the fastest method.

Methods	Performance (%)	Nb of features	Extraction time	Classification time
LBP	87.5 ± 1.3 %	243	0.34	4.6×10^{-5}
GLCM	82.6 ± 1.3 %	14	0.06	2.3×10^{-5}
SCATNET	92.5 ± 1.3 %	417	0.9	6.9×10^{-5}
CNN	95.2 ± 1.4 %	107×10^{6}	1.6×10^{-3}	

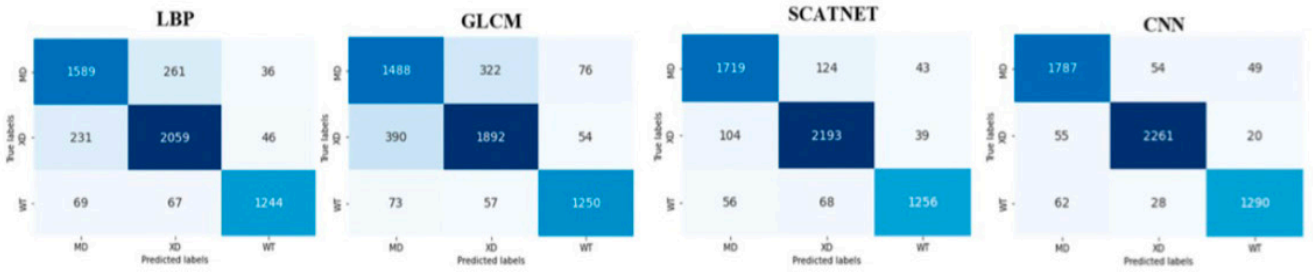

FIGURE 10 – Confusion matrices for the cells classification computed as the sum of all confusion matrices of test images during the 10−folds cross validation method for each classifical approach (LBP, GLCM and SCATNET) and deep learning CNN method.

4.3 Results and discussion

The classification performances regarding the three cell classes WT, XD, and MD, based on the textural and CNN feature spaces are presented in Table 1. Similarly, the confusion matrices showing the number of truly and falsely classified cells for each applied method are presented in Figure 10.

First, for the classical machine learning classification approach, the best cell classification performance was found when scattering tranform method is used with an accuracy of 92.5 %. This overcome the LBP and GLCM accuracies of 87.5% and 82.6% respectively. This is justified by the fact that the SCATNET extract features at a multiple cell scale level (see Figuee 5) compared to LBP and GLCM which are monoscale methods.

Second, for the deep learning approach, the classification performance outperform the classical approach with an accuracy of 95.2%. This obtained high accuracy value is due to the high-level features accessible via the CNN architecture.

In addition, as we are looking for a real time cell acquisition and classification. We compared the feature extraction and classification times for the tested strategies. These computation times are shown in Table 1 and they were obtained with an Intel Core i7-6700HQ CPU @ 2.60 GHz for the classical feature extraction approaches and with a Tesla V100-DGXS-32GB GPU for CNN deep learning approach. Although the classification efficiency is high for images with the classical SCATNET classification approach, the computation time is still not compatible with a real time application given that the needed time to extract and classify 5 cells is about 4.5 seconds, i.e. larger then 1 second needed to acquire them. On the other side, for the CNN approach, where the classification accuracy is slightly better then the SCATNET method, the classification time of one cell is very fast (1.6×10^{-3}), i.e. compatible with an application of real time screening of cells (time of 8×10^{-3} seconds for the classification of 5 cells less than 1 seconds the time needed to acquire them).

As a consequence, this comparison allows us to consider that the classification based on CNN method of cell images acquired with a cell flow speed of 140 nl/min could be suitable for real time high-throuput classification of the cells.

Conclusion

In this note, we presented an overview of machine learning and deep learning classification methods. We compared these approaches based on a real use case of multi-class classification of fluorescence microscopy cell images. Based on the results obtained for this use case, the CNN method using high multi-scale features outperforms the conventional methods with an accuracy of 95.2%. In this note, the classification tasks were performed on 2D images, it will be interesting to compare the results of the methods for 3D volume classification microscopy cell images using 3D variants of the presented methods such as LBP-TOP and 3D GLCM for the classical methods and 3D CNN for the deep learning approach.

Références

[1] Christian Janiesch, Patrick Zschech, and Kai Heinrich. Machine learning and deep learning. *Electronic Markets*, 31(3) :685–695, 2021.

[2] Mark Nixon and Alberto Aguado. *Feature extraction and image processing for computer vision*. Academic press, 2019.

Références

[3] Ayodeji Olalekan Salau and Shruti Jain. Feature extraction : a survey of the types, techniques, applications. In *2019 International Conference on Signal Processing and Communication (ICSC)*, pages 158–164. IEEE, 2019.

[4] Ali Ahmad, Carole Frindel, and David Rousseau. Detecting differences of fluorescent markers distribution in single cell microscopy : textural or pointillist feature space ? *Frontiers in Robotics and AI*, 7 :39, 2020.

[5] Majid Mirmehdi. *Handbook of texture analysis*. Imperial College Press, 2008.

[6] Robert M Haralick, Karthikeyan Shanmugam, et al. Textural features for image classification. *IEEE Transactions on Systems, Man, and Cybernetics*, 3(6) :610 – 621, 1973.

[7] Timo Ojala, Matti Pietikäinen, and Topi Mäenpää. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 24(7) :971 – 987, 2002.

[8] Joan Bruna and Stéphane Mallat. Invariant scattering convolution networks. *IEEE transactions on pattern analysis and machine intelligence*, 35(8) :1872–1886, 2013.

[9] Christopher M Bishop. *Pattern recognition and machine learning*. springer, 2006.

[10] Marti A. Hearst, Susan T Dumais, Edgar Osuna, John Platt, and Bernhard Scholkopf. Support vector machines. *IEEE Intelligent Systems and their applications*, 13(4) :18–28, 1998.

[11] Ragav Venkatesan and Baoxin Li. *Convolutional neural networks in visual computing : a concise guide*. CRC Press, 2017.

[12] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4700–4708, 2017.

[13] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv :1409.1556*, 2014.

[14] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[15] Vittoria Poli, Luca Fagnocchi, Alessandra Fasciani, Alessandro Cherubini, Stefania Mazzoleni, Sara Ferrillo, Annarita Miluzio, Gabriella Gaudioso, Valentina Vaira, Alice Turdo, et al. Myc-driven epigenetic reprogramming favors the onset of tumorigenesis by inducing a stem cell-like state. *Nature communications*, 9(1) :1–16, 2018.

[16] Federico Sala, Michele Castriotta, Petra Paiè, Andrea Farina, Sarah D'Annunzio, Alessio Zippo, Roberto Osellame, Francesca Bragheri, and Andrea Bassi. High-throughput 3d imaging of single cells with light-sheet fluorescence microscopy on chip. *Biomedical optics express*, 11(8) :4397–4407, 2020.

Nous sommes Canopee, un cabinet de conseil indépendant, spécialiste en Finance, DATA et transformation digitale.

Nous sommes l'*empowering ecosystem* ! Un écosystème en perpétuel mouvement.
Un écosystème qui donne le pouvoir à nos collaborateurs, nos projets et nos clients de se nourrir de l'émulation collective pour lui donner de la force.

Depuis 2009, nous intervenons dans les secteurs de la BFI de l'Asset Management, la banque privée ou de détail, les services financiers et l'assurance et cela auprès de clients tels que SG, HSBC,BNP Paribas ou encore ALLIANZ GI et AXA IM. Des écosystèmes réglementaires et spécifiques qui ont été nos premiers terrains de jeu.
C'est ici que nous avons su développer notre agilité, nos compétences, notre sens de l'engagement. Aujourd'hui, nous grandissons et continuons à nous investir et nous engager sur des écosystèmes spécifiques, techniques, mais diversifiés tels que l'industrie, le retail ou encore la pharma.
Toujours autour de nos 3 expertises que sont la finance, la data et la transformation digitale.

canopee
empowering *ecosystem*

www.canopee-group.com