"Innovations in Science and Technologies" scientific electronic journal
ISSN: 3030-3451
www.innoist.uz
Volume 2
№ 1
January, 2025

EFFICIENCY METHODS OF THE CNN DNN MODEL IN ENHANCING THE QUALITY OF MICROSCOPIC IMAGES AND THEIR VIZUALIZATION

Ismailov O.M,
ISFT t.f.d.,prof,
ismoilovotabek2023@gmailcom
+998 99 885 56 93

Eshmurodov D.E,
TATU t.f.n.dotsent,
e_dilshod69@mail.ru
+998977672010

Tulaganova F.K,
TATU doctoral student,
tulaganovafotima043@gmail.com
+998978007180

Abstract. *The application of deep training algorithms is of great importance when it comes to detecting micrascopic images and isolating them from objects. In particular, convolution neural networks (CNN) show high efficiency in object recognition and image analysis in medicine. This article is devoted to the application of deep training methods for segmentation and object recognition of microscopic objects and the comparative analysis of these methods with traditional and other algorithms. With CNN, the images are cut into small pieces and the main features are identified in each piece. In the process, the color values of the image (RGB) are transmitted to the network through the input layer and then analyzed using filters in the convolution layer. Cotton fibers and bacteria are separated by means of Sobel and Canny filters. This article explores the effectiveness of segmentation and classification algorithms.*

Keywords*: Deep training (Deep Learning), convolutional neural network (CNN), micrascopic image segmentation, image analysis, object separation, classification algorithms.*

1. Introdaction

Increase the quality of micrascopic images and identify objects. Quality detection of micrascopic images in many cases, object recognition processes are being implemented through manual recording methods [1]. Traditional methods involve manual processes that take a lot of time and labor, resulting in a low level of accuracy. Therefore, with the development of modern technologies, the need arose to analyze the quality of cotton fibers using automated systems[2] Today, deep training algorithms play a major role in the automatic recognition and analysis of objects in medical processes. In particular, deep training algorithms such as convolutional neural networks (CNN) have high efficiency in segmenting and classifying images, and can be used to extract and quantify objects from micrascopic images. CNN algorithms analyze images through filters to identify important features in image fragments. This ensures accurate analysis of the cotton fibers based on the images.[3].This study examines the applications and benefits of deep training algorithms for segmentation and bacterial extraction of micrascopic images. Also, the efficiency of algorithms is compared with traditional manual methods and other algorithms[4,5].

2. Works performed

A deep neural network (DNN) is an artificial neural network composed of several layers between the input and output layers. These layers can be repetitive neural network layers or convolutional layers, making DNN a more complex machine learning algorithm[6,7]. DNNs have the ability to recognize and analyze sound, creative thinking, voice commands.

DNN is a type of machine learning algorithm that learns through repeated actions from many samples. When you provide the computer with information, DNN sorts the data based on its elements, such as sound height. Data is transmitted through successive layers until it is able to accurately determine the type of sound created in the data. The model then receives feedback on the correct answer, which reinforces her learning process. Deep neural networks also have their own advantages of use[8]. DNNs between inputs and outputs. In deep learning, classification algorithms consist of different layers that work together in the process of identifying and classifying images or other data.

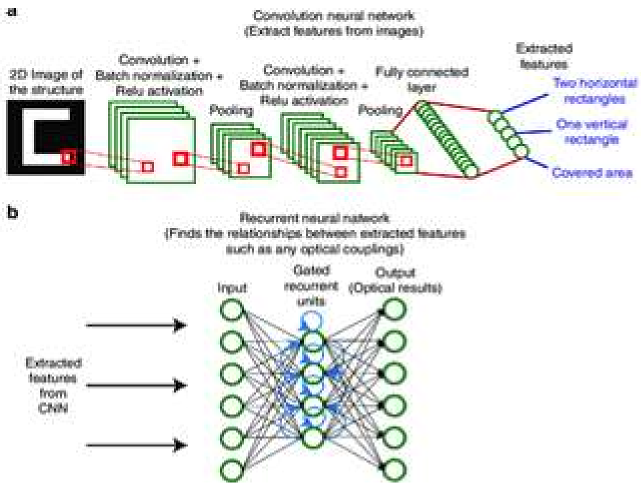

Figure 1.1 Deep training neural network performance architecture with 9 steps.

1. Input layer (Input Layer) the color values of each pixel (RGB or other color formats) enter the input layer

2. Convolutional layer (Convolutional Layer) filters (Kernel): convolutional layers divide images into small pieces (filters or cores), with each filter identifying a specific feature in the image (e.g. lines, edges).

3. The activation function (Activation Function) changes the negative values in the maps to 0 and preserves the positive values[9,10].

4. Pooling layer (Pooling Layer) Pooling layers reduce the size of the image and select the highest value (e.g. choosing the largest value from a 2x2 block

5.Normalization (Normalization) the normalization process calculates the average value and dispersion of properties and normalizes properties

6. Hidden layers (Hidden Layers) release high-level features and classification results

7. The output layers (Output Layer) classification layer calculates probabilities to mark images and selects the most likely Class.

8. Identifying values that are redundant with an error (loss) account in the redistribution (Backpropagation) classification process and updating network parameters

9. Hypermetric Tuning optimizes hypermetrics such as the learning rate of models, number of layers, filter sizes. This process helps to improve the overall performance of the model.

3. Mathematical operations

Below is what mathematical calculations can be performed on the incoming layer and how this process can be expressed in formulas

RGB color format: if the image is in RGB color format, each pixel will consist of three values: Red (R), Green (G), and Blue (B). For each pixel $({}_{i,j})$ in the incoming layer, the RGB color values are expressed as[10].

$$Pixel_{i,j} = (R_{i,j}G_{i,j}B_{i,j}) \tag{1}$$

Where, $R_{i,j}G_{i,j}$ and $B_{i,j}$ are the values in red, green, and blue, respectively. The image will have dimensions (h x w x d), where h is the height of the image, w is the width, and d is the number of color channels (e.g. for RGB on 3 channels). The incoming layer takes the image as a 3D array.

$$\text{Image}=\{Pixcel_{i,j,k}\}^{h} i = 1 - j - 1 *\underline{\ \ }^{w}{}_{k} = 1 \tag{2}$$

$$Normalized\ Pixel_{i,j,k} = \frac{Pixcel_{i,j,k} - Mi}{Max - Min} \tag{3}$$

Where is the height index, j is the width index, and k is the color channel index. Normalization: data normalization is the implementation of image pixel values to enter a certain range. For example, it can be between 0 and 1.

$$\text{Image}=\{Pixcel_{i,j,k}\}^{64} i = 1 - j - 1 *\underline{\ \ }^{64}{}_{k} = 1^{3} \tag{4}$$

Where I and j are the indices along the dimensions, K represent the color channels (1 Red, 2 Green, 3 Blue).

$$Normalized\ Pixel_{i,j,k} = \frac{Pixcel_{i,j,k}}{255} \tag{5}$$

1. The incoming layer performs the necessary mathematical operations to obtain data in digital format and correctly transfer it to the next layers. This, in turn, ensures that the data works properly in the learning process of the neural network

Using a convolution operation, a filter or kernel is applied to the image, which helps the filter identify features in the image.

Mathematical Formula: if image I, K kernel or filter and O result is an image, the convolution operation is expressed as:

$$O(_{i,j}) = \sum_{m=1}^{M} \sum_{n=1}^{N} I(i+m-1, j+n-1) * K(m,n) \qquad (6)$$

Here:
$O(_{i,j})$ result in location.
I$(i+m-1, j+n-1)$ the Pixel value of the incoming image in its location.
K(m,n)kernel elements in the location.
M and N-the dimensions of the kernel are.

2. Kernel a small size Matrix (filter) used in a convolutional layer, it helps to identify features in the image. For example, a 3 × 3 kernel.

$$K=\begin{bmatrix} k_{1,1} & k_{1,2} & k_{1,3} \\ k_{2,1} & k_{2,2} & k_{2,3} \\ k_{3,1} & k_{3,2} & k_{3,3} \end{bmatrix} \qquad (7)$$

Where k is the kernel elements.
Padding (filling) if p is a padding value, the new size of the image

$$W_{out} = \frac{W_{in}-F+2P}{S} + 1 \qquad (8)$$

Here:
W_{in} is the width of the incoming image.
F is the size of the kernel.
P-padding value.
S-stride (step).
Stride (step): if stride is S the convolution results in a new dimension

$$W_{out} = \frac{W_{in}-F+2P}{S} + 1 \qquad (9)$$

Here W_{out} is a new size.
Bias (difficulty) calculation with Bias b

$$O(_{i,j}) = \sum_{m=1}^{M} \sum_{n=1}^{N} I(i+m-1, j+n-1) * K(m,n) + b \qquad (10)$$

Where B is the bias value.

3. Activation function 3 plays a key role in neural networks because it defines the output signal of a neuron and increases the complexity of the network. Activation functions allow the output of neurons to extend beyond the line, adding a high degree of nonlinearity to the network.

Leaky ReLU Function

Leaky ReLU is a variant of the ReLU function, maintaining a small gradient for negative values.

$$Leaky\ ReLU_{(X)} = \begin{cases} x & agar\ x>0 \\ a & agar\ x<0 \end{cases} \quad (11)$$

Where α is a small positive constant (usually 0.01).

4. Main Types Of Pooling Layer

1. Max Pooling
2. Average Pooling
3. Global Pooling

In the Max pooling layer, the largest value is chosen in a small part of the image (e.g. 2x2 or 3x3). This helps to store the part of the information that has the most characteristics.

$$MaxPooling_{(Xi,j)} = \max(x_{i,j}) \quad (12)$$

Where $(x_{i,j})$are the pixel values within the pooling Window (window).

max $(x_{i,j})$ is the maximum value in the pooling window.

5. Hidden layer.The operation of the ular is the basis of neural networks, which apply the functions of transformation (activation functions) to neurons and them to carry out mathematical calculations. Neurons in hidden layers process the input through interconnected weights. A linear combination is performed using weights (weights) and biase (bias) linked by input values of each neuron:

$$Zj = \sum_{i=1}^{n} wjiXi + Bj \quad (13)$$

Where Xi is the I-repeat property in the incoming layer.

wji is the weight of neuron for I-repeat characteristic.

Bj is a Biase of neuron.

Zj is the output of neuron, a linear combination result.

6. The main processes of Hypermetric tuning when choosing Hypermetrics, the following parameters can be considered:

Learning rate (Learning Rate): determines how fast a model learns at each iteration.

Batch size (Batch Size): number of samples used to study once.

Epoch: how many iterations or epochs to train a model.

Number of layers (Number of Layers): number of layers in the Model.

Number of Neurons per Layer (): number of neurons in each layer.

Regularization parameters (Regularization Parameters): applied to avoid overfitting.

For each combination, training and evaluation of the model is carried out. Typically, grid search uses the following formula:

$$\text{Brest Parameters} = \underset{\text{Parameters}}{\arg\min}\ \text{Validation Loss}$$

Where "Validation Loss" is the model's error in the validation set.

The following is a method of segmentation of cotton fibers by filtration with the method of neural network Incretionv3 ResNet 34 VGG16.

Error functions in Deep Learning: cross-entropy error (Cross-Entropy Loss): used for classification problems.

$$Cross\ Entropy\ Loss = \sum_{c=1}^{C} Yclog(y\hat{}c) \qquad (14\ )$$

Where C is the number of classes, Yc is the real probability, $y\hat{}c$ is the estimated probability.

Error in output layer 8: the error in the re-distribution (Backpropagation) output layer $\delta^{(L)}$delta is calculated as follows:

$$\delta^{(L)^*} = y\hat{} - y \qquad (15)$$

Here:

$y\hat{}$- predicted network value,

y is the real value.

For intermediate layers, the error is calculated in the l-Layer as follows:

$$\delta^{(L)^*} = \left(W^{(L+1)}\right)T\delta^{(i+1)}0f'(z^{(l)}) \qquad (16)$$

Here:

$\delta^{(L)}$- next layer error

$W^{(L+1)}$ is the matrix of weights in the next layer, F'(z(l) is the derivative of the activation function of the

L – Layer

0 Is the Hadamard manifold (element-wise multiplication). Weights and biases are updated using the gradient descent method:

$$W\_^{(l)} = W\_^{(l)} - \eta \frac{\partial L}{\partial L\_^{(l)}} \qquad (17)$$

Here:

η-reading rate(learning rate,

$\partial L\ \partial W\ (l)$ - gradient of weights in L-Layer,

$\partial L\_^{(l)}$ is the gradient of the biases in the L-Layer.

Calculation of gradents:

$$\frac{\partial L}{\partial W\_^{(l)}} = \partial(l)(a\_^{(l-1)})T$$

$$\frac{\partial L}{\partial w\_^{(l)}} = \partial(l) \qquad (18)$$

The essence of this process is that during each training period, the network changes its weight to optimal values, which ensures high accuracy in image recognition. 3. 4.

Results.

Read deep in Table 1.2 below in business issues, a table of comparative analysis of the main algorithms used to identify and distinguish objects in images by indicators is presented.

Algorithm name	Pace	Number of adjustable parameters	Complexity	Using the Aprior parameter	Search quality (accuracy)
R-CNN region-CNN	Low	Middle (millions)	High Medial	Small	High(but works slowly)
FastR-CNN	Middle	Middle	Middle High	Small	High
Fastr R-CNN	High	Middle	Medial	Small	High(fast and high quality)
YOLO	Too high	Low (millions)	Medial	Minemal	Middle
SSD	High	Middle	Medial	Minemal	Middle High
Mask R-CNN	Low	Medial High (millions of parameters)	Medial	Minemal	Veryhigh(with sigmenting)
Retina Net	Middle	less	High	Small	High (but low)
Efficient Dent	High	Less	Medial	Minemal	High

Table 4.1 comparative analysis of object recognition and separation algorithms in images used in solving deep training issues.

In the table, the YOLO (you Only Look Once) algorithm is distinguished by efficiency. This algorithm is capable of detecting objects at high speed in real time. The number of parameters is also relatively low, as well as the fact that it analyzes images at once makes it effective in terms of resources and time. Especially in real-time issues (such as automatic traffic control or security systems), YOLO works very well. This algorithm is fast and relatively high in accuracy, as it only looks at the entire image once at each training stage. The R-CNN (Region-CNN) algorithm can be seen as the most inefficient. Although this algorithm gives good results in accuracy, its speed is very low. The reason is, dividing each image into segments and then studying each segment separately increases the time and resource demand. Adjusting many parameters also increases complexity, making it difficult to use in real time.Below are the results obtained by the method of the neural network

Incretionv3ResNet34VGG16.4.

Algorithm name	Speed (FPS)	Accuracy (Map %)	Error Rate (%)
InceptionV3 YOLO	30	85	5
ResNet 34 YOLO	40	90	10
VGG16 YOLO	20	80	15

Table 4.2 results obtained by Incretionv3 ResNet 34 VGG16 neural network method are given.

According to the speed and error rate of accuracy of the resulting roughness, the quality of cotton fibers is performed efficiently in the ResNet 34 YOLO model, which is able to plan them effectively in a short time.

Conclusion

This study examined the application of CNN and DNN algorithms to image recognition in micpascopic image segmentation and object separation. The results showed that CNN performed faster and more accurately than traditional methods. It has also been observed that CNN makes effective use of resources and has high results in real-time issues in medicine. In the future, it is planned to work on optimizing network architecture and hypermetric parameters to improve the efficiency of deep Training Networks.

Reference

1. Majumdar, A. Majumdar, PK, Sarkar, B. Anwendung eines adaptiven Neurofuzzy-Systems zur Vorhersage der Festigkeit von Baumwollfäden aus den Eigenschaften von HVI-Fasern. Zeitschrift des Textilinstituts, Band 96 (1), S. 55-60, 2008.
2. Chauhan., Yadav, N., Arya, N. künstliche neuronale Netzwerkanwendungen in Textilien. Internationale Zeitschrift für aktuelle Mikrobiologie und angewandte Wissenschaften, Band 7 (4), S. 3134-3143, 2018.
3. Ghosh, A., Hasnath, A., Halder, S. Es wurde ein wahrscheinliches neuronales Netzwerk vorgeschlagen, um Baumwollfadendefekte unter Verwendung eines Systems zu klassifizieren. Jüngste Erfolge und Innovationen im Ingenieurwesen (ICRAIE), 2014.
4. Rahadi, I., Choodoung, M. Und Choodoung, A. (Mai 2020). Identifizierung von roten Blutkörperchen und weißen Blutkörperchen durch Bildverarbeitung. In Zeitschrift für Physik: eine Reihe von Konferenzen (Vol. 1539, Nr. 1, S. 012025). In: IOP publishing.
5. Harms, H., Aus, H. M., Haucke, M., & Gunzer, U. (1986). Segmentation of stained blood cell images measured at high scanning density with high magnification and high numerical aperture optics. Cytometry: The Journal of the International Society for Analytical Cytology, 7(6), 522-531.

6. Bai, R., Jiang, S., Sun, H., Yang, Y., & Li, G. (2021). Deep neural network-based semantic segmentation of microvascular decompression images. Sensors, 21(4), 1167.

8. Eshmuradov D.E Tulaganova F.K Biological composition of blood cells and diagnosis of patients with pathological diseases, INTERNATIONAL CONFERENCE ON RESEARCH IN SCIENCES, EDUCATION AND HUMANITIES p 101-106 28.02.2024

9. Eshmurodov D.E Iskanderova S.N Tulaganova F.K ALGORITHMS FOR DETECTION OF CELLS IN BLOOD IMAGES, INNOVATIVE RESEARCH AND PERSPECTIVES OF THE DEVELOPMENT OF SCIENCE AND TECHNOLOGY Innovative research and perspectives of THE development of science and technology p 328-331 29.01.2024

10. Saleem, S., Amin, J., Sharif, M., Mallah, G.A., Kadry, S., & Gandomi, A.H. (2022). Leukemia segmentation and classification: A comprehensive survey. Computers in Biology and Medicine, 150, 106028.