An efficient local binary pattern based plantar pressure optical sensor image classification using convolutional neural networks

Article

Accepted Version

It is advisable to refer to the publisher's version if you intend to cite from the work. See Guidance on citing.

To link to this article DOI: http://dx.doi.org/10.1016/j.ijleo.2019.02.109

www.reading.ac.uk/centaur

CentAUR

Central Archive at the University of Reading

Reading's research outputs online

Full-Text version

Title:
An efficient local binary pattern based plantar pressure optical sensor image classification using convolutional neural networks

Authors:
Cunlei Wang[a,b], Donghui Li[a], Zairan Li[c], Dan Wang[c], Nilanjan Dey[d], Anjan Biswas [e,f,g], Luminita Moraru[h], R. S. Sherratt[i], and Fuqian Shi[j]

[a] School of Electrical Automation and Information Engineering, Tianjin University, Tianjin, 300072, P. R China
[b] Tianjin Vocational College of Mechanics and Electricity, Tianjin, 300350, P. R. China
[c] Tianjin Key Laboratory of Process Measurement and Control, School of Electrical Engineering and Automation, Tianjin University, 300072, P.R. China
[d] Dept. of IT, Techno India College of Technology, West Bengal, 740000, India
[e] Department of Physics, Chemistry and Mathematics, Alabama A&M University, Normal, AL 35762, USA
[f] Department of Mathematics and Statistics, Tshwane University of Technology, Pretoria 0008, South Africa
[g] Department of Mathematics, King Abdulaziz University, Jeddah-21589, Saudi Arabia
[h] Faculty of Sciences and Environment, Department of Chemistry, Physics and Environment, Dunarea de Jos University of Galati, 47 Domneasca Str.,800008, Romania
[i] Department of Biomedical Engineering, the University of Reading, RG6 6AY, UK
[j] College of Information & Engineering, Wenzhou Medical University, Wenzhou, 325035, P.R. China

Corresponding author:
Fuqian Shi, #7B-219, Chashan Campus, Wenzhou Medical University, Wenzhou, 325035, P.R. China, Tel. +86-577-86689913; Fax. +86-577-86699222, E-mail: sfq@wmu.edu.cn.

Abstract:
The objective of this study was to design and produce highly comfortable shoe products guided by a plantar pressure imaging data-set. Previous studies have focused on the geometric measurement on the size of the plantar, while in this research a plantar pressure optical imaging data-set based classification technology has been developed. In this paper, an improved local binary pattern (LBP) algorithm is used to extract texture-based features and recognize patterns from the data-set. A calculating model of plantar pressure imaging feature area is established subsequently. The data-set is classified by a neural network to guide the generation of various shoe-last surfaces. Firstly, the local binary mode is improved to adapt to the pressure imaging data-set, and the texture-based feature calculation is fully used to accurately generate the feature point set; hereafter, the plantar pressure imaging feature point set is then used to guide the design of last free surface forming. In the presented experiments of plantar imaging, multi-dimensional texture-based features and improved LBP features have been found by a convolution neural network (CNN), and compared with a 21-input-3-output two-layer perceptual neural network. Three feet types are investigated in the experiment, being flatfoot (F) referring to the lack of a normal arch, or arch collapse, Talipes Equinovarus (TE), being the front part of the foot is adduction, calcaneus varus, plantar flexion, or Achilles tendon contracture and Normal (N). This research has achieved an 82% accuracy rate with 10 hidden-layers CNN of rotation invariance LBP (RI-LBP) algorithm using 21 texture-based features by comparing other deep learning methods presented in the literature.

1. Introduction

Pressure sensors are the most commonly used sensor in industrial practice, and the pressure sensor primarily uses the piezoelectric effect [1] [2]. The photoelectric sensor can be used to detect the non-electric physical quantities that directly cause the changes of optical quantities, such as light intensity, illumination, radiation temperature measurement, gas composition analysis, etc. They can also be used to detect other non-electric quantities that can be converted into light changes, such as the diameter of parts, surface roughness, strain, displacement, vibration, speed, acceleration, and the shape of objects, the identification of working state, etc. [3]. By collecting the dynamic pressure distribution of the sole, the biomechanical properties of shoes can be evaluated, such as cushioning performance, support strength, stability and so on [4] [5]. By previous works, the results indicated that foot zoning, pressure, press, risk assessment of diabetic foot ulcer, load change rate, foot length and width, foot contact area, foot angle, foot axis, and time track are used for stability analysis [6][7]. It is an important direction of this research to use sole pressure imaging data set and segmentation results to control the bottom surface modeling, and then improve the wearing comfort of shoes [8]. Menz et al. [9] compared the plantar pressure between healthy individuals of three feet status as normal, planus and cavus, and confirmed each foot posture classification displays unique plantar pressure characteristics through a statistics feature of plantar image feature- "peak pressures". Rome et al. [10] illustrated the relationship between region-specific foot pain and plantar pressure and concluded that foot pain at the midfoot and rearfoot is relative to the increasing plantar pressure of toes.

Through the measurement of plantar pressure distribution and gait analysis, scientific shoe selection suggestions can be provided for athletes to avoid unnecessary injury risk caused by shoe discomfort. At the same time, personalized correction programs can be formulated based on stress test data to improve comfort, enhance endurance, improve competitive level and prevent sports injury. Murphy et al. [11] tested many kinds of shoes to show the plantar pressure's importance for the research. Otherwise, two test data can be compared and analyzed: pressure distribution comparison, synchronization comparison, impulse comparison, force curve comparison, stability comparison, cerebral palsy pressure distribution comparison, preoperative and postoperative gait comparison; multiple test data can be averaged: pressure average, valgus average; provide personalized correction program [12] [13] [14].

Image classification is a new science and technology developed in recent years. Classification methods based on image space mainly use the underlying features of image such as color, gray level, texture, shape and position to classify images. Image classification is based on texture feature, which describes the distribution characteristics of image gray space [15]. Texture is everywhere, and there are different textures on the surface of each object. The texture of the image can be processed by computer and digitized, and then the signal of image analysis and processing can be obtained. In the 1970s, Haralick [16] proposed a gray level co-occurrence matrix representation method. This method defines the gray level co-occurrence matrix by using the joint probability density of gray level pixels in two locations, reflecting the direction, adjacent interval, change range of the gray level of the image, and thus obtains the texture feature vector. However, due to the diversity of texture images and the complexity of analysis algorithms, it is difficult to apply the same texture feature in different fields [17]. Traditionally, texture is a macroscopic representation of a local repetitive pattern of eigenvalue intensity in an image, and local pattern repetition and stability are its main characteristics. Texture image classification refers to the designation of a predefined texture category based on the content of the image or image region to be classified; it is widely used in visual navigation and image classification. Scene classification, object recognition, face recognition, intelligent video analysis, content-based image and video retrieval, remote sensing image analysis, industrial detection, medical image analysis and text classification, as depicted in Figure 1.
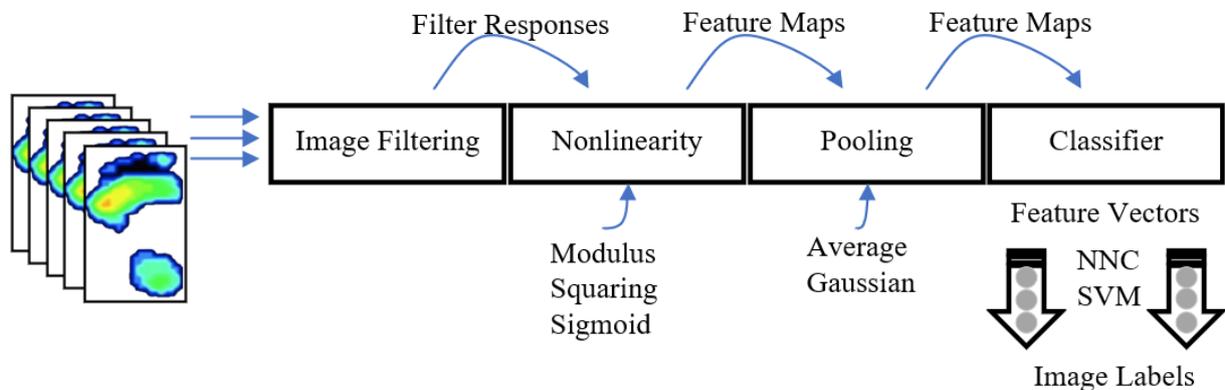

Figure 1. General model of texture-based image classification

For the analysis and understanding of various natural image textures in the actual open environment, innovative research on theory and algorithm is still needed to better learn and express texture features, and at the same time to improve the theoretical analysis, promote. Texture feature extraction method is widely used in practical engineering, among which the important issues to be further studied and solved include the construction of large-scale texture database [18].

Deep convolution neural network for texture analysis and understanding; efficient classification of texture images; robust texture classification in open environment; semantic understanding of texture images; and many potential applications for two-dimensional texture analysis. For example, industrial surface inspection, remote monitoring, biopharmaceutical image analysis and other fields [19]. But in practical applications the texture is not as regular as in experiments, and there are many other issues including: illumination changes caused by non-uniform light source; the direction of the objects is random; the spatial scale is inconsistent. Moreover, many texture classification methods are too complex to be practical. To solve these problems, an improved local two value pattern can be used for texture classification. Ojala et al. [20] generalized gray-scale and rotation invariant operator presentation to present a method for combining multiple operators for multiresolution analysis.

The improved Local Binary Patterns has the following advantages: gray scale invariance; rotation invariance; multi-resolution analysis; and, because of its low computational complexity, it is a very practical image processing method [21] [22]. Some LBP based improvement algorithms have also been introduced, Yuan et al. [23] proposed Gabor filter and CS-LBP operator to extract the polarized image texture effectively. Yang et al. [24] introduced histogram-based LBP for face recognition by comparing with naïve holistic LBP and enhanced LBP. Öztürk et al. [25] deployed an image classification using LBP and texture features. Otherwise, fractal-based texture classification [26], dictionary learning with LBP [27], Histogram of Oriented Gradient (HOG) based LBP [28] were also introduced subsequently.

From image extraction to feature expression, an image can be described by a fixed dimension vector, and then a classifier is learned to classify the image. The commonly used classifiers are SVM, nearest neighbor, neural network, random forest, etc. Convolutional Neural Networks/CNNs/ConvNets are very similar to traditional neural networks as they are all composed of neurons with learnable weights and biases. Each neuron receives some input and does some point product calculation. The output is the score of each classification. Some computing techniques in ordinary neural networks are still applicable here. The CNN is specially designed for image recognition. It imitates the multi-layer process of human recognition image: pupil ingestion of pixels; some cells in the cerebral cortex preliminary processing to find shape edges and directions; Abstract judgment of shape (such as circle and square); further Abstract judgment [29]. The difference between convolution neural network and a traditional neural network is that convolution neural network contains a feature extractor composed of convolution layer and sub-sampling layer. In convolution layer of convolution neural network, a neuron is connected only with some adjacent layer neurons. In a convolution layer of CNN, there are usually several feature maps. Each feature plane is composed of some rectangular neurons. The neurons in the same feature plane share weights. The shared weights here are the convolution core. Convolution kernels are usually initialized in the form of random decimal matrices. In the process of network training, the convolution kernels will learn to obtain reasonable weights. The direct benefit of shared weights (convolution kernels) is to reduce the connection between layers of the network, while reducing the risk of over-fitting. Sub-sampling is also called pooling. There are usually two forms: mean sub-sampling and Max sub-sampling. Subsampling can be regarded as a special convolution process. Convolution and subsampling greatly simplify the model complexity and reduce the parameter [30] of the model. Paoletti et al. [31] presented a new 3D CNN architecture for classifying hyperspectral images effectively. Wan et al. [32] compared AlexNet, VggNet, GoogleNet, and ResNet for diabetic retinopathy image classification.

This paper introduces a novel LBP texture-based classification using a CNN for plantar pressure imaging dataset which directed from optical sensor acquisition device provides a guidance for shoe customization and has a potential application in shoe production industry to create comfortable shoes for people with a wide variety of medical conditions or sports enhancement. The organization of the remaining sections is as follows. Section 2 introduces the proposed methods. Section 3 addresses the results and comparing analysis. Section 4 involves the conclusions of the present studies.

2. Methods

2.1 Feature extraction and selection

In the texture feature description method, the typical representation based on statistical method is a texture feature analysis method called gray level co-occurrence matrix. Gotlieb and Kreyszig et al. [33] based on the study of various statistical features in the co-occurrence matrix, through experiments, obtained four key features of gray level co-occurrence matrix: energy, inertia, entropy and correlation. Another typical statistical method is to extract texture features from image autocorrelation function (i.e. image energy spectrum function), that is, to extract texture parameters such as texture thickness and orientation by calculating image energy spectrum function. The so-called geometric method is based on texture primitives (basic texture elements). A texture feature analysis method based on the theory of texture primitives.

According to the theory of texture primitives, a complex texture can be composed of several simple texture primitives arranged repeatedly in a regular form. In geometric methods, there are two more influential algorithms: Voronio checkerboard feature method and structural method. The model method is based on the image construction model. Typical methods are random field model method, such as Markov random field (MRF) model method and Gibbs random field model method. Signal processing method: Texture feature extraction and matching mainly include gray level co-occurrence matrix, Tamura texture feature, autoregressive texture model and wavelet transform. Gray level co-occurrence matrix feature extraction and matching mainly depend on four parameters: energy, inertia, entropy and correlation. Tamura texture feature is based on human visual perception psychology of texture. Six attributes are proposed: roughness, contrast, direction, line image, regularity and roughness. Autoregressive texture model (simulate) Neous Auto-Regressive (SAR) is an application example of Markov random field (MRF) model.

2.2 Texture feature calculation using statistics methods

2.2.1 Gray difference-based features

Suppose that (x, y) is a point's coordinates in plantar pressure image, $x \in \{1, 2, ..., m\}$, $y \in \{1, 2, ..., n\}$, gray level L, and the nearby point is $(x + \Delta x, y + \Delta y)$, $\Delta x > 0$, $\Delta y > 0$, $f(x, y)$ is the value of point. So, the gray value can be calculated by,

$$g_\Delta(f(x, y)) = g(f(x, y)) - g(f(x + \Delta x, y + \Delta y)) \tag{1}$$

where, g_Δ is gray difference, and suppose there are m levels, count g_Δ in each m and the probability on each g_Δ can be acquired as $p(i)$, $(i = 1, \cdots, m)$,

$$mean = \frac{1}{m} \sum_{i=1}^{m} i p(i) \tag{2}$$

$$contract = \sum_{i=0}^{m} i^2 p(i) \tag{3}$$

$$entropy = -\sum_{i=0}^{m} p(i) \log_2(p(i)) \tag{4}$$

2.2.2 Gray-level Co-occurrence Matrix based features

Energy, correlation and inertia moment are defined as,

$$energy = \sum_{x=0}^{m-1} \sum_{y=0}^{m-1} P_{xy}^2 \tag{5}$$

where, P_{xy} means the probability of point (x, y) in $i = 1, \cdots, m$ level.

$$correlation = \frac{1}{\sigma_X \sigma_Y} \sum_{x=0}^{m-1} \sum_{y=0}^{m-1} (x - \mu_X)(y - \mu_Y) P_{xy} \tag{6}$$

In which

$$\mu_X = \sum_{x=0}^{m-1} x \sum_{y=0}^{m-1} p_{xy}, \mu_Y = \sum_{y=0}^{m-1} y \sum_{x=0}^{m-1} p_{xy}, \sigma_x = \sqrt{\sum_{x=0}^{m-1} (x - \mu_X)^2 \sum_{y=0}^{m-1} p_{xy}}, \sigma_y = \sqrt{\sum_{y=0}^{m-1} (y - \mu_Y)^2 \sum_{x=0}^{m-1} p_{xy}}.$$

$$im = \sum_{i=0}^{L-1} \sum_{j=0}^{L-1} |i - j|^k P_{ij} \tag{7}$$

2.2.3 Gray gradient co-occurrence matrix

Let gray image be $g(x, y)$; apply Sobel operator to $f(x, y)$ and discrete $g(x, y)$ to $G(x, y)$; let new gray level be L_g, hence, the new gray is calculated as,

$$G(x, y) = \frac{g(x, y) - \min(g(x, y)}{\max(g(x, y)) - \min(g(x, y))}(L_g - 1) \tag{8}$$

GGCM H_{ij} is defined as number of $\{(x, y) \mid f(x, y) = o, G(x, y) = j\}$. Normalized H_{ij} as,

$$P_{ij} = \frac{H_{ij}}{\sum\limits_{i=0}^{L-1}\sum\limits_{j=0}^{L-1} H_{ij}} \tag{9}$$

Now we defined fifteen features based on P_{ij}, continuously, we have the T1-T15 calculations in Appendix (I).

2.3 LBP features

2.3.1 LBP definitions

Local binary pattern is an operator used to describe local texture features of images; it has significant advantages such as rotation invariance and gray invariance. The original LBP operator is defined as a window of 3x3. The gray value of eight adjacent pixels is compared with the threshold value of the central pixel of the window. If the surrounding pixel value is larger than the central pixel value, the position of the pixel is marked as 1, otherwise 0. In this way, eight points in a 3x3 neighborhood can be compared to produce eight binary digits (usually converted to decimal digits, i.e. LBP codes, totaling 256). The LBP value of the central pixel of the window can be obtained and used to reflect the texture information of the area. As shown in Figure 2.
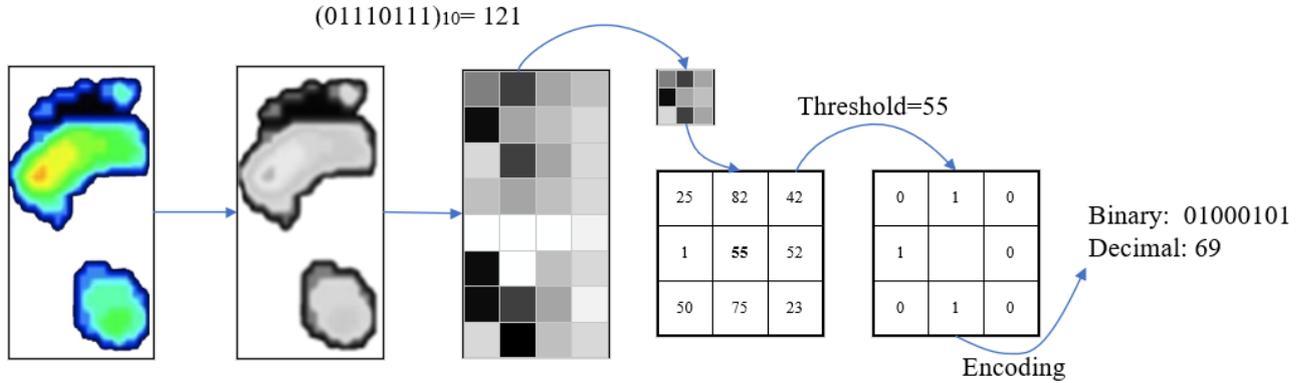

Figure 2. local binary pattern process

There have three main steps for LBP algorithm as firstly, is to calculate the binary values of neighbor pixels (0 or 1) comparing with central pixel; and secondly is to encode the binary numbers into local structure pattern, continuously to decimal, finally, histogram of the image is presented for those defined textures. Let $s(x) = \begin{cases} 1 & x \geq 1 \\ 0 & x < 0 \end{cases}$, we have that,

$$LBR_P^R = \begin{cases} \sum\limits_{i=0}^{P-1} s(g_i - g_c) & if \ U(LBP) \leq 2 \\ P+1 & otherwise \end{cases} \tag{10}$$

Where, g_i denotes gray value of each neighbor pixel of a circle with R radius, P is total numbers pixels, g_c means central pixel.

2.3.2 Efficient LBPs

As Eqn. (10), a completed LBP was defined as

$$LBP_P^R = 2^P \sum\limits_{i=0}^{P-1} s(\mid g_i - g_c \mid -c) \tag{11}$$

Where, c is a threshold (mean value of $\mid g_i - g_c \mid$). Another improvement is local binary counting methods (LBC, applied for 1 only) by [34] to encode the local distribution as shown in Figure 3.
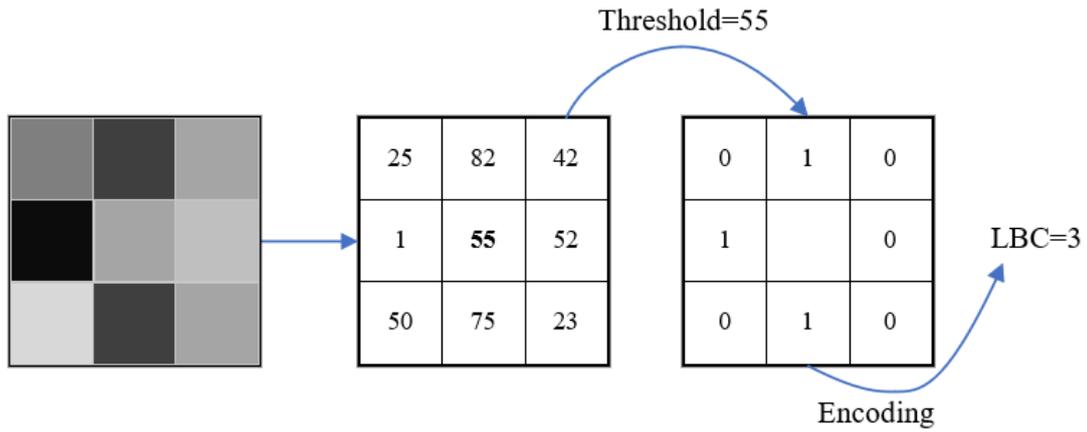
Figure 3. Local binary count model

2.3.3 Proposed LBP for optical sensor imaging dataset

(1) Circular LBP operator

The biggest drawback of the basic LBP operator is that it only covers a small area within a fixed radius, which obviously cannot meet the needs of different sizes and frequency textures. To adapt to texture features of different scales and meet the requirements of gray scale and rotation invariance, Ojala et al. [20] improved the LBP operator by extending the 3x3 neighborhood to any neighborhood and replacing the square neighborhood with the circular neighborhood. The improved LBP operator allows any number of pixels in the circular neighborhood with radius R. The LBP operator with P sampling points in a circular region with radius R is obtained, which is called Extended LBP or Circular LBP. Using the radius of the circle to code adjacent pixels, the following neighbors can be obtained as depicted in Figure 4.
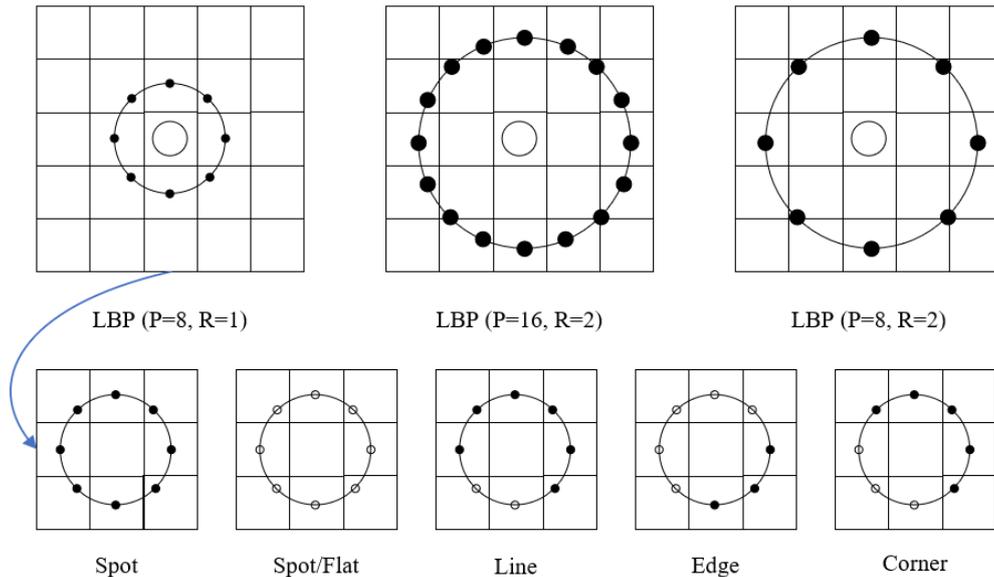
Figure 4. Extended LBPs and their neighbor pixels encoding

For a given center point (x_c, y_c), its neighborhood pixel position is (x_p, y_p), $p < P$, and its sampling point (x_p, y_p) is calculated by the following formula,

$$(x_p, y_p) = (x_c + R\cos(2\pi p / p), y_c - R\sin(2\pi p / p)) \tag{12}$$

Where R is the sampling radius, p is the sampling point, and P is the sampling number. Since the calculated value may not be an integer, that is, the calculated point is not on the image, we use the interpolation point of the calculated point. There are many interpolation methods for purposes. we use bilinear interpolation. The formula of bilinear interpolation is as follows,

$$f(x, y) \doteq \begin{bmatrix} 1-x, & x \end{bmatrix} \begin{bmatrix} f(0,0) & f(0,1) \\ f(1,0) & f(1,1) \end{bmatrix} \begin{bmatrix} 1-y \\ y \end{bmatrix} \qquad (12)$$

(2) LBP with rotation invariant mode

From the definition of LBP, we can see that the LBP operator is grayscale invariant, but it is not rotationally invariant. The rotation of the image will give different LBP values. Mäenpää et al. [35] extended the LBP operator and proposed a rotation invariant LBP operator (RI-LBP), i.e. a series of initial defined LBP values were obtained by continuously rotating the circular neighborhood, and the minimum value was taken as the LBP value of the neighborhood. Figure 5 shows the process diagram of finding rotation-invariant LBP. The number below the operator in the figure represents the corresponding LBP value of the operator. The eight LBP modes shown in the figure are processed by rotation-invariant, and the final LBP value with rotation-invariant is 15. The rotation invariant LBP patterns corresponding to the 8 LBP patterns in Figure 5 are 00001111.
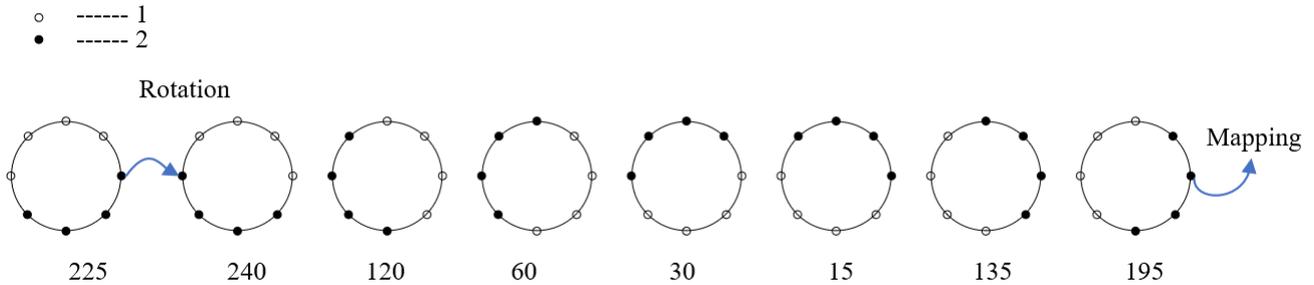

Figure 5. rotation invariant mode

(3) LBP equivalence model

An LBP operator can generate different binary modes. For a circular region with radius R, an LBP operator with P sampling points will generate two modes. Obviously, with the increase of the number of sampling points in the neighborhood set, the number of binary patterns increases dramatically. For example, there are 20 sampling points in a 5x5 neighborhood, and 2^{20}=1048756 binary patterns. So many binary patterns are disadvantageous to texture extraction, texture recognition, classification and information access. At the same time, too many pattern types are unfavorable to the expression of texture. For example, when LBP operator is used in texture classification or face recognition, the statistical histogram of LBP pattern is often used to express the information of image, and more types of patterns will make the amount of data too large and the histogram too sparse. Therefore, it is necessary to reduce the dimension of the original LBP mode to make the best representation of image information when the amount of data is reduced. To solve the problem of too many binary modes and improve the statistics, Ojala et al. [20] proposed a "Uniform Pattern" to reduce the dimension of LBP operators where they believed that in real images, most LBP modes contain at most two jumps from 1 to 0, or from 0 to 1. Therefore, Ojala et al. defined ab "equivalent schema" as an equivalent schema class when the cyclic binary number corresponding to an LBP jumps at most twice from 0 to 1 or from 1 to 0 For example, 00000000 (0 jumps), 000000111 (including only one jump from 0 to 1), 10001111 (jumping from 1 to 0, then from 0 to 1, a total of two jumps) are equivalent schema classes. Schemas other than equivalent schema classes fall into another category, called hybrid schema classes, such as 10010111 (four jumps in total). With this improvement, the number of binary patterns is greatly reduced without losing any information. The number of patterns is reduced from the original 2^P to $P(P-1)+2$, where P represents the number of sampling points in the neighborhood set. For 8 sampling points in a 3x3 neighborhood, the binary mode is reduced from 256 to 58, which makes the dimension of eigenvectors less and reduces the impact of high frequency noise.

The principle of LBP feature for detection: obviously, the extracted LBP operator can get an LBP "coding" at each pixel. After extracting the original LBP operator from an image (recording the gray value of each pixel), the original LBP feature is still a "picture" (recording). The LBP value of each pixel. In the application of LBP, such as texture classification and face analysis, LBP atlas are not usually used as feature vectors for classification and recognition, but statistical histogram of LBP feature spectrum is used as feature vectors for classification and recognition. Because, from the above analysis, we can see that this "feature" is closely related to the location information. If we extract these "features" from two pictures directly and make discriminant analysis, it will cause great errors because of "misalignment of position". Later, researchers found that a picture can be divided into several sub-regions. LBP features can be extracted for each pixel in each sub-region, and then a statistical histogram of LBP features can be built in each sub-region. In this way, each sub-region can be described by a statistical histogram; the whole picture is composed of several statistical histograms; for example, a 100x100-pixel image is divided into 10x10=100 sub-regions (which can be divided in many ways), and each sub-region is 10x10-pixel in size. In each sub-region, each pixel extracts its LBP features, and then establishes a statistical

histogram. Thus, the picture has 10x10 sub-regions, and 10x10 statistical histograms. Using this 10x10 statistical histogram, the picture can be described. Then, we can use various similarity measure functions to judge the similarity between the two images; the steps of extracting LBP feature vectors:

Step 1: first divide the detection window into 16 x 16 small area (cell).

Step 2: For one pixel in each cell, the gray value of eight adjacent pixels is compared with it. If the surrounding pixel value is larger than the central pixel value, the position of the pixel is marked as 1, otherwise 0. In this way, 8-bit binary digits can be generated by comparing 8 points in a 3x3 neighborhood, that is, the LBP value of the central pixel of the window can be obtained.

Step 3: Then calculate the histogram of each cell, that is, the frequency of occurrence of each number (assumed to be the decimal number LBP value); then normalize the histogram by Euclidian distance.

Step 4: Finally, the statistical histogram of each cell is connected to a feature vector, which is the LBP texture feature vector of the whole image.

Then it can be classified by using machine learning algorithms.

ALGORITHM 1: Rotation Invariant of LBP features
REQUIRED

img: source plantar pressure images,
    r: radius of circle
    n: neighbors

imgSize ← size(img)
IF numel(imgSize) > 2
    imgG← rgb2gray(img) // **Convert color system from RGB to Gray**
ELSE
    imgG ← img
ENDIF
[rows, cols] ← size(imgG); rows←int16(rows); cols←nt16(cols)
imglbp ← uint8(zeros (rows-2*radius, cols-2*radius))
FOR k **IN** [0, n-1]
//   **Calculate the offset of the sampling point to the center point coordinate, rx and ry**
    rx ← r * cos (2.0 * pi * k / neighbors); ry ← -r * sin (2.0 * pi * k / neighbors);
//   **The sampling points are shifted up and down respectively.**
    x1 ← floor(rx); x2 ← ceil(rx); y1 ← floor(ry); y2 ← ceil(ry);
//   **Mapping coordinates offset to [0-1].**
    tx ← rx - x1; ty ← ry - y1
//   **The weight is calculated according to the weight formula of X and y between 0-1. The weight is independent of the specific position of coordinates and the difference between coordinates.**
    w1 ← (1-tx) * (1-ty); w2 ← tx * (1-ty); w3 ← (1-tx) * ty; w4 ← tx * ty;
    FOR i **IN** [radius+1, rows-r]
        FOR j **IN** [r+1, cols-r]
          Center← imgG(i, j)
//   **According to the bilinear interpolation formula, we calculate the gray value of the K sampling points.**
          n ← imgG(i+x1, j+y1)*w1 + imgG(i+x1, j+y2)*w2 + imgG(i+x2, j+y1)*w3 + imgG(i+x2, j+y2)*w4;
//   **The LBP value of each neighbor of the LBP feature image is accumulated by operation, and the corresponding LBP value is obtained by shifting.**
          IF n > center **THEN**
            flag = 1
          ELSE
            flag = 0
          ENDIF
          imglbp(i-r, j-r) ← bitor(imglbp(i-r, j-r), bitshift(flag, n-k-1))
        ENDFOR
    ENDFOR
ENDFOR
FOR i **IN** [1, rows-2*r]
    FOR j **IN** [1, cols-2*r]
        currentValue ← imglbp(i, j)
        minValue ← currentValue
        currentValue ←= dec2bin(currentValue)
        FOR k **IN** [1, n]
          temp ← circshift(currentValue, k)
          temp ← bin2dec(temp);
          IF temp < minValue **THEN**
            minValue = temp;
          ENDIF
        ENDFOR
        imglbp(i, j) ← minValue;
    ENDFOR
ENDFOR

2.4 Proposed convolutional neural network model for optical sensor imaging dataset

A full-convolution neural network (FCN) is proposed to generate dense classification (pixel-by-pixel classification), which is characterized by position-independent and output is the result of a series of convolution layers. Firstly, the full connection layer responsible for classification is transformed into a convolution layer, and the convolution kernel is selected to make its dimension the same as the previous layer, so that it is equivalent to the full connection layer. The difference is that when the size of the input image increases, the output size will increase, but the number of parameters will not change. This is convoluting the entire original network over a larger range of images to obtain output at different locations. To improve the resolution of the output graph, we added a so-called "deconvolution" layer. The purpose of this method is to sample the feature map of the previous layer and implement it through neighborhood point interpolation. The interpolation process is parameterized by a core representing the influence area and contribution of pixels to its neighborhood. To effectively interpolate, the core must be large enough to cover the output. The implementation of interpolation is as follows: multiplying the value of the kernel with each input and adding overlapping responses to the output. As shown below, a fixed 4x4 core is used for scaling.
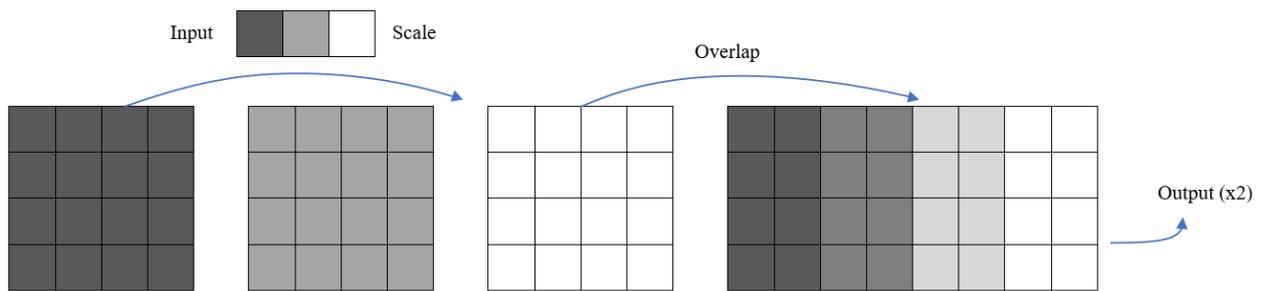

Figure 6. Scaling and overlapping

In this algorithm, the interpolation kernel is also a parameter to be learned, not an *a priori* parameter (for example, set to linear interpolation). Note that the central part of the feature map after sampling is the superposition (interpolation) of two neighborhood kernels, and the outer boundary is the result (extrapolation) of a single core. Compared with the block-based method, the advantages of this method are as follows: 1) it can eliminate the discontinuity of block boundary; 2) it can simplify the learning process and improve the accuracy with fewer parameters; 3) it has lower reasoning time and uses GPU to accelerate convolution processing. The FCN we propose is constructed as follows. Firstly, we assume that the output block size of the original neural network is 1x1, so we only need to pay attention to a single output centered on the receptive field. In the second step, we modify the full connection layer into a convolution layer containing a feature map, and the spatial dimension of the former layer is 9x9. In the third step, we add a deconvolution layer with a factor of 4 (we need to learn 8x8 kernels) to restore the input resolution. Note that the classification and the up-sampling process are separate.
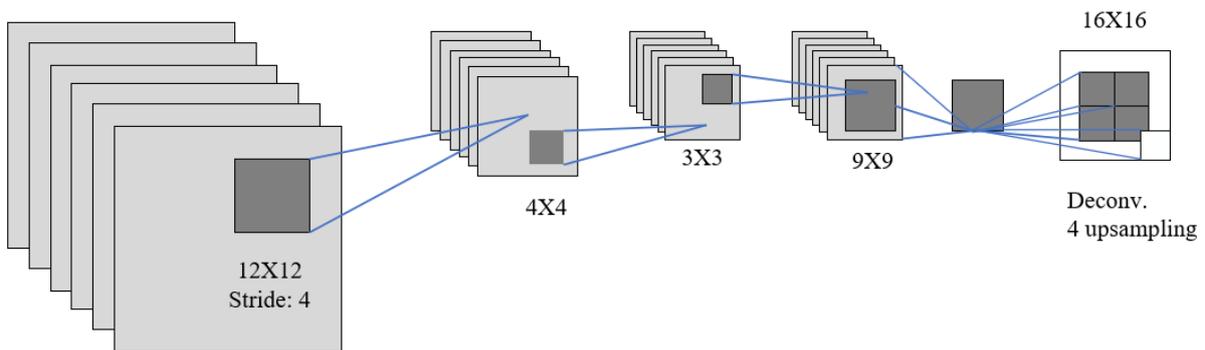
Figure 7. Deconvolution and up-sampling process

This new neural network can accept input images of different sizes, and the output size will change accordingly. In the above figure, the extrapolation part of the deconvolution result is expressed in white, and the interpolation part is expressed in grayscale. To modify the influence of mislabeled samples, we propose a two-stage algorithm: first, the original data set is used to train the neural network, and secondly, a small number of artificial labeled samples are used to optimize the neural network. Due to the loss of spatial information caused by the down sampling, the classification boundary is blurred. To preserve spatial details without increasing the number of training samples, we combine low-resolution remote features with high-resolution local features. Fine-Tuning is a common technique of neural networks, which is used to update a trained model on a new training set. The principle of Fine-Tuning is that low-level information

/ features can be reused in different applications. Although the final classification objectives are different, it is also a feasible scheme to obtain approximate optimal parameters, rather than random initialization. After Fine-Tuning, low-level features will tend to be retained while high-level parameters will be updated. In Fine-Tuning, the training set in the new domain is generally much smaller than that in training the original neural network. This is because we assume that the commonalities of the two domains are preserved in the network (e.g. edges in different directions). When the training set of new fields is very small, additional considerations need to be added to Fine-Tuning to avoid over-fitting, such as early stopping, fixing low-level weights, or reducing learning rate. Combining the idea of Fine-Tuning, to train on imperfect data, the first step is to train a complete CNN with many training samples to capture the generality of data sets, such as obtaining spectral characteristics of object classes. In the second step, we use a small number of artificially labelled samples to Fine-Tune neural network. If most of the general features have been obtained in the initial training stage, the optimization step should modify the network parameters locally to produce more accurate output. Resolution optimization directly using down-sampled CNN for classification will result in lower classification resolution, but we do not need full resolution filters, because we only need to get high resolution information of the convolution filter center (a pile of analysis). Combined with these two scales, the positioning accuracy can be improved under the condition that few parameters are maintained. Let S be the fractional order of the original resolution, such as $S = \{1, 1/2\}$ consisting of the original resolution and its half resolution. We let X_s indicate that X is down-sampled to S-level. We design a specific type of neuron, which will apply a set of filters to different scales of the previous feature map to increase response, and we have that,

$$a = \sigma(\sum_{s \in \Re} W_S X_S + b) \tag{13}$$

Note that each filter W_s learns from each scale s. Such a filter can easily be realized through the combination of basic convolution, down sampling and upper sampling level. The following figure shows the situation of the process in two scale models. In our implementation, we average down-sampled neighborhood elements in a window, then perform bilinear interpolation for up-sampled (other methods are also feasible). The convolution kernels for each scale are set to be the same (e.g. 3x3), but the context is different (because the scales are different). Addition is an element by element operation followed by nonlinear activation.
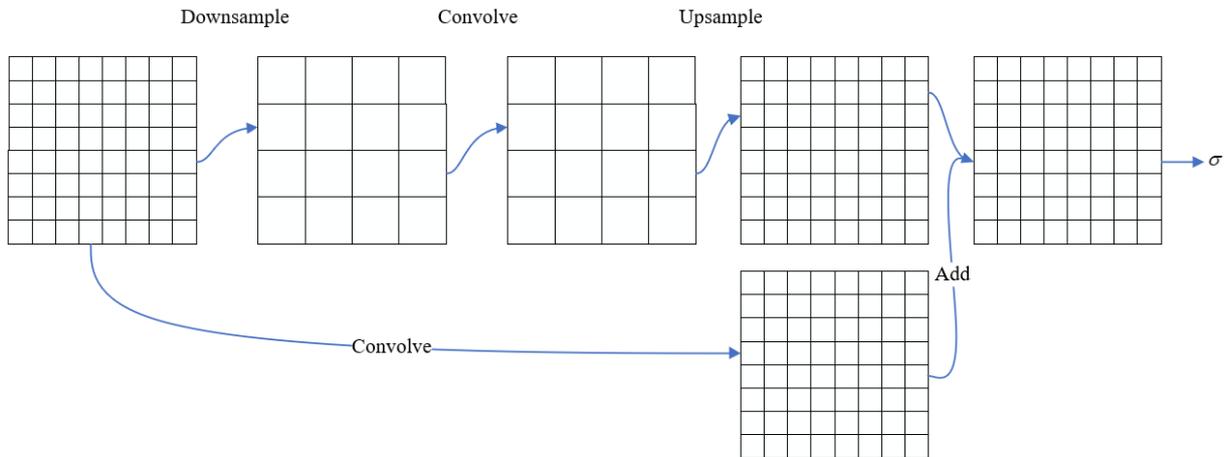
Figure 8. Convolutional neural network classification for optical sensor imaging dataset

3. Results and discussion

3.1 Experimental design and data acquisition

Table 1 shows the details of the parameters of the scanning device in the proposed system while acquiring the planar pressure distribution imaging dataset using the RS-Scan Footscan 7.0 software system (as shown in Fig.9). Twenty testee (5 Flat foot (F), 5 Talipes Equinovarus (TE) [36] and 10 normal (N)) were involved in the thirty minutes of four times test, and the data was acquired subsequently.

Table 1. The parameters of plantar pressure measurement system

Items	Measurement	Items	Measurement
Area	40*50 (cm^2)	Total number of sensors	4096
Numbers of sensor	4 /cm^2	Sampling (Hz)	125-300Hz
Size of each sensor	0.5*0.7(cm^2)	Analog channels	16
Image resolution	12 bits	Entry level	2m

Furthermore, the system can also calculate the pressure information in different zone (shown in Figure. 9). The system can not only measure the gait assessment related parameters, such as stride length, stride length, cadence, speed, feet and angle, steering angle and other data, but also through the intelligent terminal feedback foot movements and plantar pressure distribution evolution diagram. Plantar pressure gait analysis assessment reports the runner distribution, human orientation, internal and external rotation, and running posture provides scientific basis for gait training and exercise guidance. The medical personnel can real-time monitoring of the patient's foot by providing comprehensive rehabilitation training suggestions.
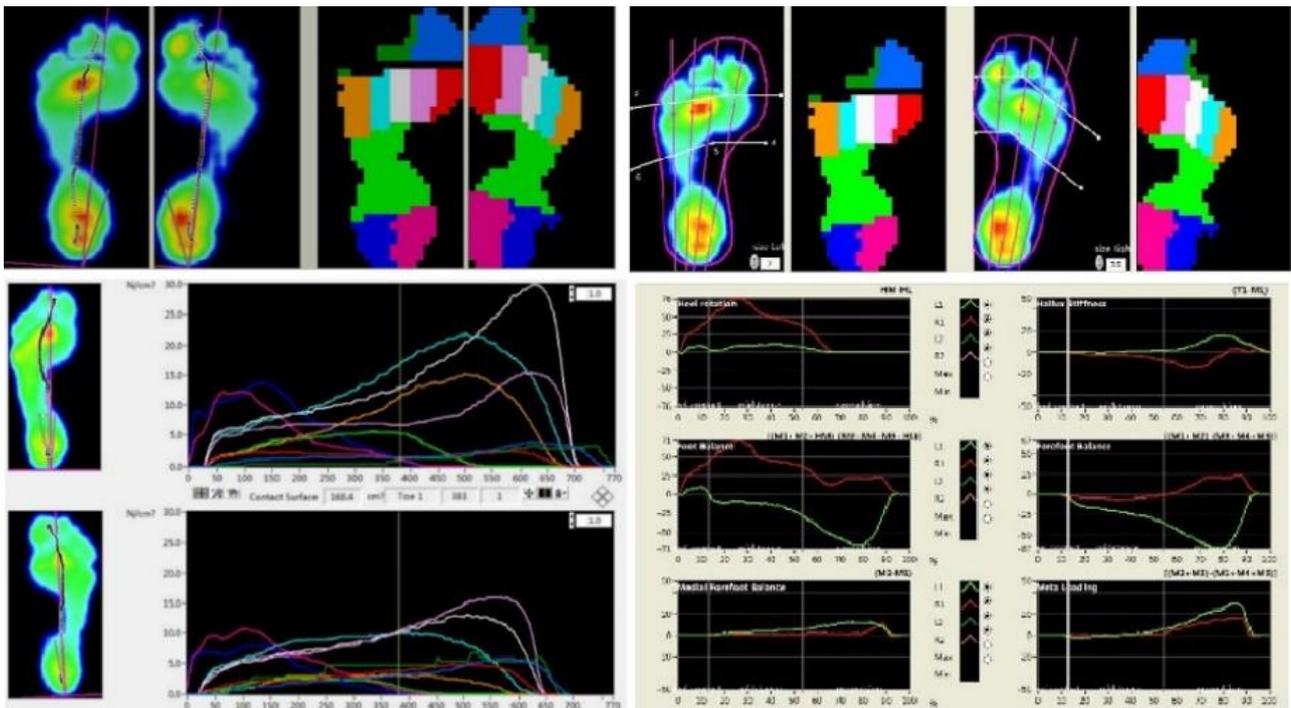
Figure 9. Foot scan system for measuring plantar pressure and imaging experiment

3.2 Results and discussion

Flat foot refers to the lack of normal arch, or arch collapse which defined in the research as Flatfoot(F); Talipes Equinovarus (TE) means the front part of the foot is adduction or varus, calcaneus varus, plantar flexion, and Achilles tendon contracture, etc. Texture features based efficient LBP, RI-LBP were calculated in Figures 11-12.

(a)   With noisy   (b) Pre-processed
Figure 11. With noisy (a) and preprocess (b) (original, efficient LBP and RI-LBP)

(a) Pulse sensors image   (b) Back remove preprocess
Figure 12. with pulse sensors image (a) and back remove preprocess (b) (original, efficient LBP and RI-LBP)
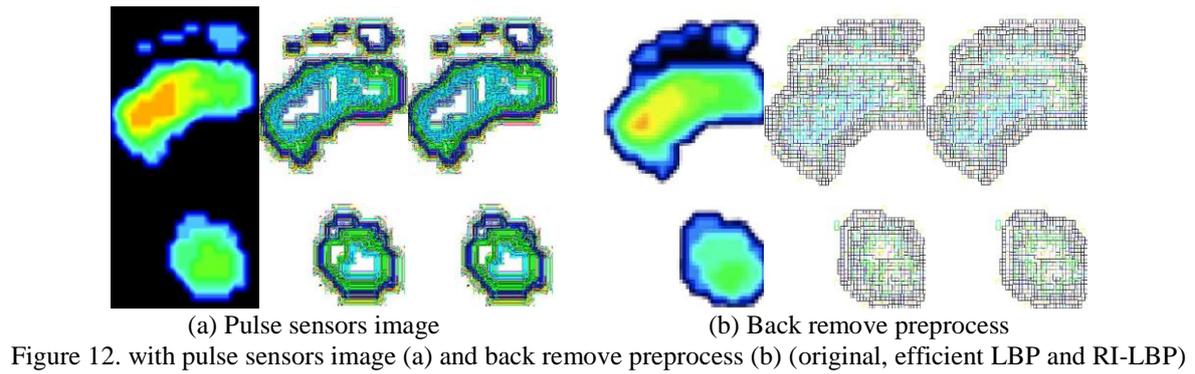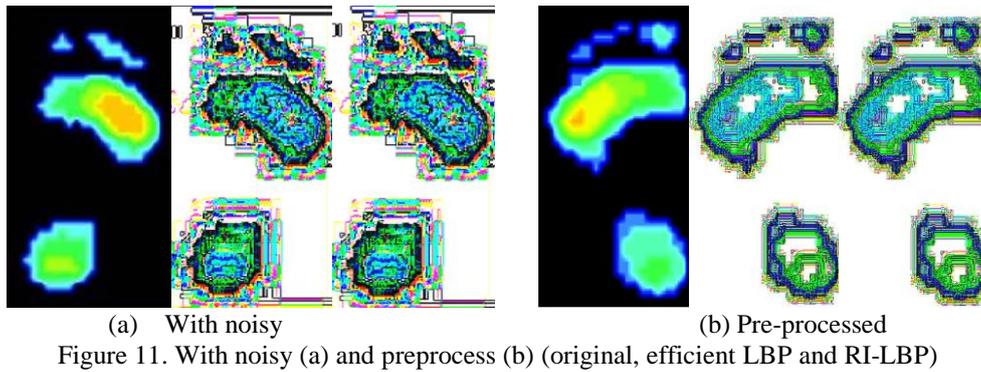
We also calculated the histogram of Figure 12-(b) as shown in Figure 13.
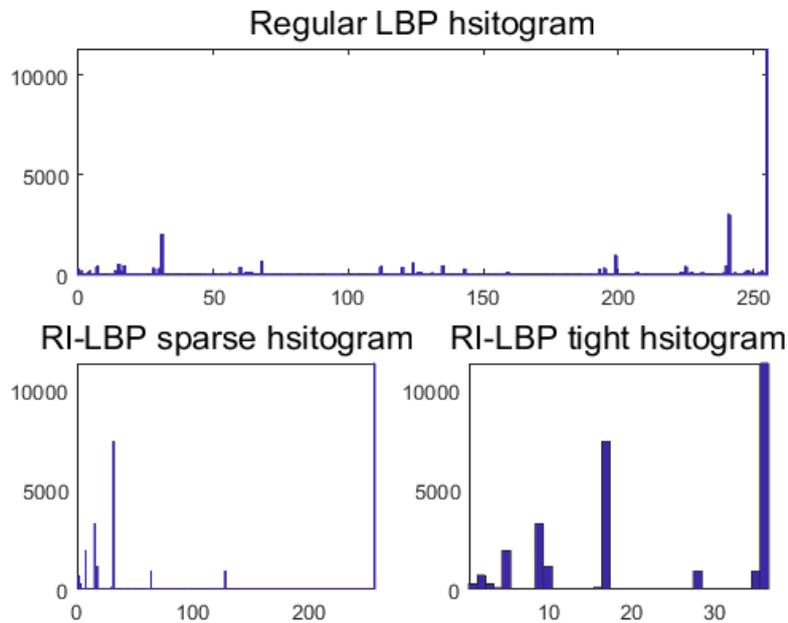
Figure 13. The regular, sparse and tight RI-LB the plantar pressure images

Normal (N) Flat foot (F), Talipes Equinovarus (TE); CNN image classification, by using CIFAR-10 (The CIFAR-10 data set contains 60,000, 32x32 color images of 10 classes, each with 6,000 images. There are 50,000 training images and 10,000 test images in the CIFAR-10 data set) for training the CNN for prediction of plantar pressure images classification. and we have the parameters as shown in Table 2.

Table 2. The parameters and sizes of convolutional neural network for plantar pressure imaging dataset training

Parameters	Size	Parameters	Size
Input	3*32*32	Convolutional layer	20*28*28
Element wise layer	20*28*28	Pooling layer	20*14*14
Convolutional layer	50*10*10	Element wise layer	50*10*10
Pooling layer	50*5*5	Flatten layer	1250
Dot plus layer	500	Element wise layer	500
Dot plus layer	10	Softmax layer	10
Output	10	-	-

Using the 10 hidden two layers feed -forward neural network with "sigmod" input and "softmax" output neurons, see Figure 14.

Figure 14. the structure of ten hidden feed-forward neural network

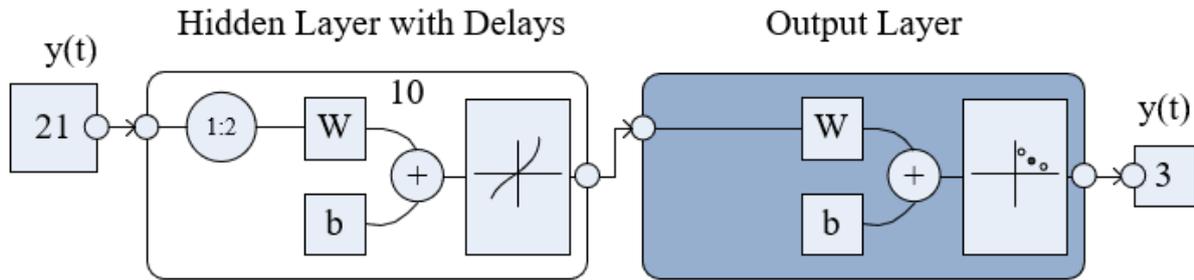T1-T15 with 6 features of mean, contract, entropy, energy, correlation and inertia moment are applied as inputs (See Table 3-7). The input is a 21x20 matrix consisting of 20 testee; the target is 3 of 1-Normal (N), 2-Flat foot (F), 3-Talipes Equinovarus (TE). Classes are represented by a 1 in row 1, 2 or 3; 80% of the dataset is for training, 10% for validation and 10% for testing. The training parameters are of random division, scaled conjugate gradient, and cross entropy for performance. The performance, training state, confusion and receiver operating characteristic are shown in Fig. 15. By using other neural network mothods of back-propagation neural network (BPNN) [37], convolutional neural network (CNN)[38], back-propagation Hopfield neural network (BPHNN) [39] and fuzzy neural network (FNN) [40], the proposed CNN with RI-LBP using 21 texture features deployed better performance on CE, MSE and PAR.

Table 3. Texture features' value of three classifications (F, TE, N)

Images	Mean (M)	Contract (Co)	Entropy (E)	Energy (En)	Correlation (C)	Inertia Moment (IM)
F1	0.58	2.37E+3	4.76	0.03	0.78	1.21
F2	0.54	2.12E+3	4.65	0.03	0.50	2.25
F3	0.57	2.21 E+3	4.93	0.02	0.54	1.57
F4	0.54	2.17 E+3	4.78	0.03	0.87	2.21
F5	0.55	2.14 E+3	4.67	0.03	0.89	1.24
TE1	0.55	2.65 E+3	4.90	0.02	0.88	1.23
TE2	0.59	2.23 E+3	4.56	0.02	0.84	1.37
TE3	0.66	2.52 E+3	5.20	0.03	0.91	1.67
TE4	0.59	2.27 E+3	5.11	0.02	0.91	2.56
TE5	0.50	2.03 E+3	5.12	0.03	0.92	2.44
N1	0.57	2.21 E+3	5.33	0.02	0.92	2.43
N2	0.51	1.98 E+3	5.20	0.03	0.91	2.21
N3	0.55	1.99 E+3	4.45	0.03	0.89	2.42
N4	0.55	1.23 E+3	4.77	0.02	0.87	1.56
N5	0.56	1.98 E+3	5.23	0.03	0.79	2.44
N6	0.57	2.05 E+3	5.22	0.02	0.83	1.88
N7	0.57	2.28 E+3	5.34	0.03	0.88	1.87
N8	0.55	2.43 E+3	5.32	0.03	0.90	1.45
N9	0.55	2.02 E+3	4.88	0.03	0.95	2.44
N10	0.53	2.21 E+3	4.87	0.03	0.91	2.88

Table 4. Texture features of Flatfoot (F)

T\F	F1	F2	F3	F4	F5
T1	0.24	0.28	0.22	0.27	0.30
T2	14.45	15.44	13.32	11.23	13.26
T3	4.88E+4	4.31E+4	5.34E+4	4.51E+4	6.83E+4
T4	2.11E+5	2.12+5	1.56+5	1.33+5	1.55E+5
T5	0.01	0.01	0.01	0.01	0.01
T6	115.33	122.21	99.41	105.70	107.13
T7	13.67	14.89	12.76	11.44	13.21
T8	55.65	54.97	45.75	53.32	69.34
T9	12.89	13.54	12.91	12.32	16.33
T10	-113.23	-114.129	-48.52	-53.23	-134.31
T11	2.23	2.80	2.65	2.89	2.67
T12	0.87	0.96	1.56	1.65	1.21
T13	3.21	3.18	3.22	3.54	3.33
T14	1.26E+4	1.32E+4	1.29E+4	1.30E+4	1.45E+4
T15	0.02	0.01	0.01	0.01	0.01

Tables 5. Texture features of Talipes Equinovarus (TE)

T\TE	TE1	TE2	TE3	TE4	TE5
T1	0.31	0.30	0.30	0.30	0.31
T2	13.25	12.23	12.33	13.52	13.56
T3	4.65E+4	4.76 E+4	4.98 E+4	4.98 E+4	5.88 E+4
T4	1.56E+5	1.57 E+5	1.52 E+5	1.51 E+5	1.59E+5
T5	0.01	0.01	0.01	0.01	0.01
T6	112.76	107.78	112.54	122.68	117.20
T7	13.45	13.20	13.55	13.45	13.34
T8	61.20	60.12	61.25	61.34	61.64
T9	11.88	11.75	11.89	11.25	11.62
T10	-55.22	-47.25	-42.22	-57.45	-89.36
T11	2.23	2.12	2.65	2.54	2.44
T12	1.12	1.07	1.05	1.05	1.09
T13	3.22	3.18	3.54	3.55	3.34
T14	1.24E+4	1.33 E+4	1.25 E+4	1.10 E+4	1.23 E+4
T15	0.01	0.01	0.02	0.01	0.01

Table 6. Texture features of Talipes equinovarus (N)

T\N	N1	N2	N3	N4	N5
T1	0.31	0.31	0.30	0.30	0.31
T2	17.87	17.22	18.32	18.22	18.39
T3	9.23 E+3	9.21 E+3	9.29 E+3	8.98 E+3	8.70E+3
T4	4.23E+5 E+5	3.23 E+5	4.33 E+5	4.66E+5	4.22 E+5
T5	0.01	0.02	0.01	0.01	0.01
T6	112.22	113.32	113.44	111.12	120.09
T7	17.23	17.12	16.87	16.22	16.22
T8	60.12	61.22	61.23	62.23	57.21
T9	12.98	12.99	12.78	13.12	13.11
T10	-134.39	-187.22	-199.02	-126.64	-187.65
T11	2.23	2.23	2.27	2.34	2.09
T12	0.83	0.82	0.81	0.79	0.77
T13	3.12	3.22	2.98	3.09	3.02
T14	1.37 E+4	1.36 E+4	1.54 E+4	1.05 E+4	1.43E+4 E+4
T15	0.01	0.01	0.01	0.01	0.01

Table 7. Texture features of Talipes equinovarus (N) (continued)

T	N6	N7	N8	N9	N10
T1	0.32	0.31	0.32	0.31	0.33
T2	15.57	16.28	16.52	16.54	16.54
T3	8.21E+3	9.23 E+3	9.12E+3	8.32 E+3	7.99 E+3
T4	3.44E+5	3.32 E+5	4.10 E+5	4.09 E+5	4.01 E+5
T5	0.01	0.01	0.01	0.01	0.01
T6	89.23	89.41	89.88	92.20	101.22
T7	16.60	16.67	16.32	16.17	16.43
T8	52.22	52.23	56.87	54.76	53.67
T9	13.21	12.12	12.55	12.57	12.22
T10	-122.12	-111.22	-117.43	-119.76	-113.85
T11	2.37	2.29	2.32	2.32	2.29
T12	0.85	0.87	0.82	0.83	0.82
T13	3.19	3.21	3.12	3.15	3.18
T14	1.09E+4	1.08 E+4	1.08 E+4	1.12 E+4	1.28 E+4
T15	0.01	0.01	0.01	0.02	0.01

(1) performance   (2) epochs

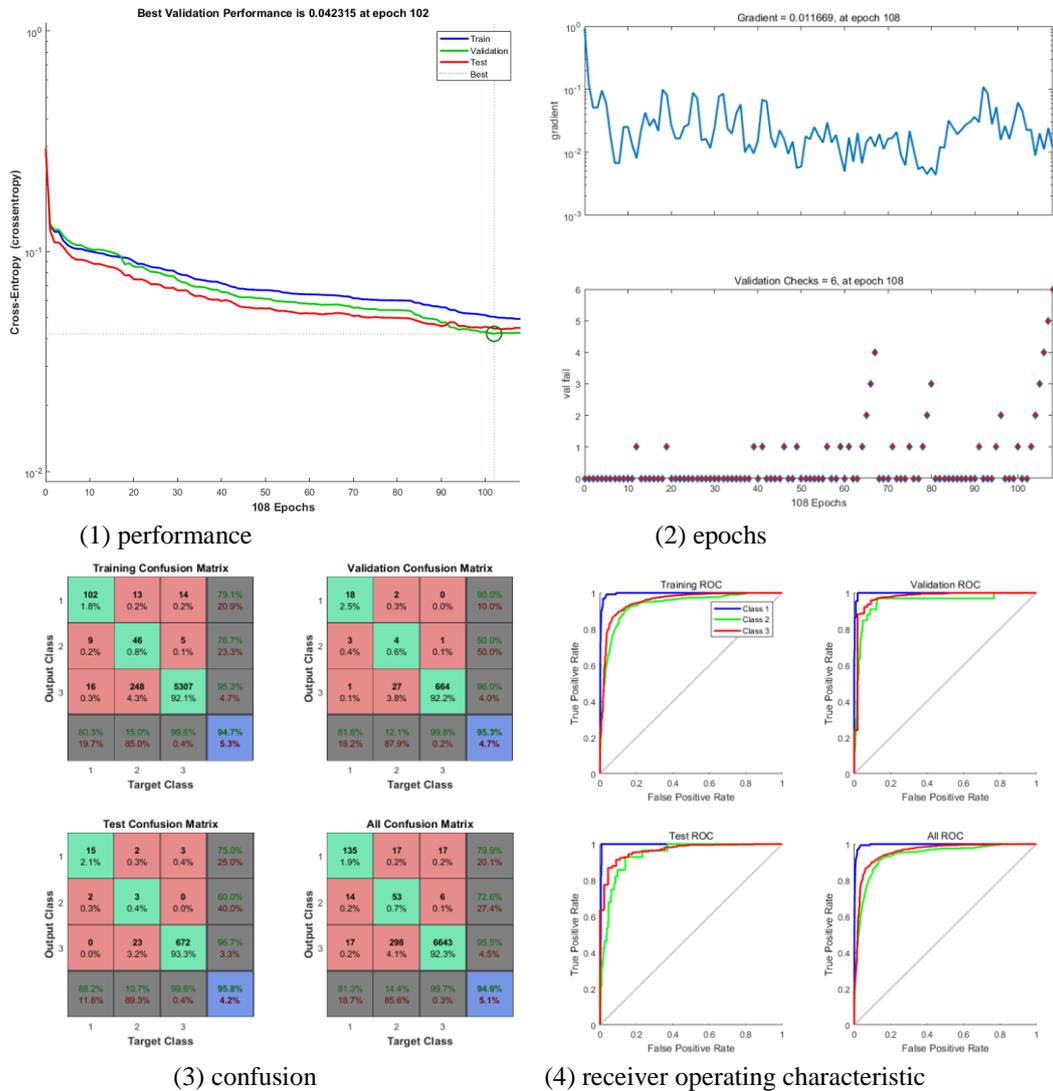(3) confusion   (4) receiver operating characteristic

Figure 15. Results on performance, training state, confusion and receiver operating characteristic

Table 8. Predictive accuracy rate with different neural networks

Methods	Cross Entropy (CE)	Mean Square Error (MSE)	Predictive Accuracy Rate (PAR)
BPNN [33]	0.62	0.077	0.855
CNN [34]	0.71	0.068	0.788
BPHNN [35]	0.61	0.066	0.685
FNN [36]	0.65	0.082	0.752
CNN + RI-LBP -21(+)	0.67	0.066	0.820

+: proposed in this research; PAR: predictive accuracy rate

4. Conclusion

In this research, plantar pressure experimental design was addressed to discover how the last can be made through the plantar pressure imaging dataset, optical sensor based imaging dataset was specified using image based process such texture feature extraction, as the emergence of the deep neural network modeling technologies, this research achieved better performance by using CNN and RI-LBP, the results showed the training set for acquiring higher classification with higher accuracy for prediction by comparing with other machine learning algorithms. The next work in future is extending the dataset to large scale and using more features. Texture feature is only a feature of the surface of an object, it cannot fully reflect the intrinsic properties of the object, so it is impossible to obtain high-level image content only by using texture features. The research based on plantar pressure data set can be further applied to the study of healthy gait by analyzing human body structure, function and whole-body posture control. In clinical application, it can be further developed from basic research related to rehabilitation using plantar pressure for dynamic image analysis. Medical studies have also shown that diabetes, obesity, Parkinson's disease and other diseases can lead to changes in plantar pressure. This will bring new research prospects and become a new driving force for the development of traditional shoemaking industry.

Compliance with ethical standards

Conflict of interest: none.

References

[1] John F. Drazan, Omar T. Abdoun, Michael T. Wassick, Reena Dahle, Luke Beardslee, George A. Marcus, Nathaniel C. Cady, Eric H. Ledet, Simple implantable wireless sensor platform to measure pressure and force, Medical Engineering & Physics, Volume 59, 2018, Pages 81-87, https://doi.org/10.1016/j.medengphy.2018.06.006.

[2] A.R. Chaudhury, A.K. Pantazis, N. Chronis, An image contrast-based pressure sensor, Sensors and Actuators A: Physical, Volume 245, 2016, Pages 63-67, https://doi.org/10.1016/j.sna.2016.04.057.

[3] Gong Wang, Tao Liu, Xiang-Chao Sun, Pu Li, Yi-Shi Xu, Jian-Guan Hua, Yan-Hao Yu, Shun-Xin Li, Yun-Zhi Dai, Xin-Yu Song, Chao Lv, Hong Xia, Flexible pressure sensor based on PVDF nanofiber, Sensors and Actuators A: Physical, Volume 280, 2018, Pages 319-325, https://doi.org/10.1016/j.sna.2018.07.057.

[4] D. Bhatta, M. Marti Villalba, C.L. Johnson, G.D. Emmerson, N.P. Ferris, D.P. King, C.R. Lowe, Rapid Detection of Foot-and-Mouth Disease Virus with Optical Microchip Sensors, Procedia Chemistry, Volume 6, 2012, Pages 2-10, https://doi.org/10.1016/j.proche.2012.10.124.

[5] Zairan Li, Nilanjan Dey, Amira S. Ashour, Luying Cao, Yu Wang, Dan Wang, Pamela McCauley, Valentina E. Balas, Kai Shi, and Fuqian Shi, Convolutional neural network based clustering and manifold learning method for diabetic plantar pressure imaging dataset, Journal of Medical Imaging and Health Informatics, 7, 3, pages 639-652, https://doi:10.1166/jmihi.2017.2082

[6] Luying Cao, Nilanjan Dey, Amira S. Ashour, Simon Fong, R.S. Sherratt, Lijun Wu, and Fuqian Shi, Diabetic plantar pressure analysis using image fusion, Multimedia Tools and Applications. 2018, doi: https://doi.org/10.1007/s11042-018-6269-x

[7] Cunlei Wang, Zairan Li, Nilanjan Dey, Amira S. Ashour, Simon Fong, R.S. Sherratt, Lijun Wu and Fuqian Shi, Histogram of oriented gradient based plantar pressure image feature extraction and classification employing fuzzy support vector machine. Journal of Medical Imaging and Health Informatics, Vlolume 8, Issue 4, Pages 842-854, doi: https://doi.org/10.1166/jmihi.2018.2310

[8] Dan Wang, Zairan Li, Nilanjan Dey, Amira S. Ashour, R.S. Sherratt, and Fuqian Shi, Case-based reasoning for product style construction and fuzzy analytic hierarchy process evaluation modeling using consumers linguistic variables, IEEE Access, Volume 5, 2017, Pages 4900-4912, doi: https://doi.org/10.1109/ACCESS.2017.2677950

[9] Andrew K. Buldt, Saeed Forghany, Karl B. Landorf, Pazit Levinger, George S. Murley, Hylton B. Menz, Foot posture is associated with plantar pressure during gait: A comparison of normal, planus and cavus feet, Gait & Posture, Volume 62, 2018, Pages 235-240, https://doi.org/10.1016/j.gaitpost.2018.03.005.

[10] Sarah Stewart, Matthew Carroll, Angela Brenton-Rule, Monique Keys, Libby Bell, Nicola Dalbeth, Keith Rome, Region-specific foot pain and plantar pressure in people with rheumatoid arthritis: A cross-sectional study, Clinical Biomechanics, Volume 55, 2018, Pages 14-17, ISSN 0268-0033, https://doi.org/10.1016/j.clinbiomech.2018.04.002.

[11] Earnest P.S. Mawusi, 20 - Shoes and Shoe Modifications, Editor(s): Joseph B. Webster, Douglas P. Murphy, Atlas of Orthoses and Assistive Devices (Fifth Edition), 2019, Pages 229-232.e1, https://doi.org/10.1016/B978-0-323-48323-0.00020-2.

[12] Renato R. Azevedo, Emmanuel S. da Rocha, Pedro S. Franco, Felipe P. Carpes, Plantar pressure asymmetry and risk of stress injuries in the foot of young soccer players, Physical Therapy in Sport, Volume 24, 2017, Pages 39-43, https://doi.org/10.1016/j.ptsp.2016.10.001.

[13] Danielle M. Torp, Abbey C. Thomas, Luke Donovan, External feedback during walking improves measures of plantar pressure in individuals with chronic ankle instability, Gait & Posture, Volume 67, 2019, Pages 236-241, https://doi.org/10.1016/j.gaitpost.2018.10.023.

[14] Jocelyn F. Hafer, Mark W. Lenhoff, Jinsup Song, Joanne M. Jordan, Marian T. Hannan, Howard J. Hillstrom, Reliability of plantar pressure platforms, Gait & Posture, Volume 38, Issue 3, 2013, Pages 544-548, ISSN 0966-6362, https://doi.org/10.1016/j.gaitpost.2013.01.028.

[15] Zhong-Qiu Zhao, Shou-Tao Xu, Dian Liu, Wei-Dong Tian, Zhi-Da Jiang, A Review of Image Set Classification, Neurocomputing, 2018, https://doi.org/10.1016/j.neucom.2018.09.090.

[16] Robert M. Haralick, Statistical and structural approaches to texture, *Proc. IEEE*, vol. 67, no. 5, pp. 786-804, 1979, https:// 10.1109/PROC.1979.11328

[17] Bibicu, D., Moraru, L., Biswas, Thyroid nodule recognition based on feature selection and pixel classification methods, Journal of Digital Imaging, Volume 26, Issue 1, 2013, Pages 119-128, http://www.mijst.mju.ac.th/vol10/220-232.pdf

[18] Avik Hati, Subhasis Chaudhuri, Rajbabu Velmurugan, An image texture insensitive method for saliency detection, Journal of Visual Communication and Image Representation, Volume 43, 2017, Pages 212-226, https://doi.org/10.1016/j.jvcir.2017.01.007.

[19] Yian Seo, Kyung-shik Shin, Hierarchical convolutional neural networks for fashion image classification, Expert Systems with Applications, Volume 116, 2019, Pages 328-339, ISSN 0957-4174, https://doi.org/10.1016/j.eswa.2018.09.022.

[20] Timo Ojala, Matti Pietikäinen, Topi Mäenpää, Multiresolution Gray-Scale and Rotation Invariant Texture Classification with Local Binary Patterns, IEEE Transactions on Pattern Analysis and Machine Intelligence, Volume 4, Issue 7, Pages 971-987, https://doi.org/10.1109/TPAMI.2002.1017623

[21] Peizhong Liu, Jing-Ming Guo, Kosin Chamnongthai, Heri Prasetyo, Fusion of color histogram and LBP-based features for texture image retrieval and classification, Information Sciences, Volume 390, 2017, Pages 95-111, https://doi.org/10.1016/j.ins.2017.01.025.

[22] Vivek H. Mahale, Mouad M.H. Ali, Pravin L. Yannawar, Ashok T. Gaikwad, Image Inconsistency Detection Using Local Binary Pattern (LBP), Procedia Computer Science, Volume 115, 2017, Pages 501-508, https://doi.org/10.1016/j.procs.2017.09.097.

[23] Baohong Yuan, Baihua Xia, Dexiang Zhang, Polarization Image Texture Feature Extraction Algorithm Based on CS-LBP Operator, Procedia Computer Science, Volume 131, 2018, Pages 295-301, https://doi.org/10.1016/j.procs.2018.04.167.

[24] Bo Yang, Songcan Chen, A comparative study on local binary pattern (LBP) based face recognition: LBP histogram versus LBP image, Neurocomputing, Volume 120, 2013, Pages 365-379, https://doi.org/10.1016/j.neucom.2012.10.032.

[25] Şaban Öztürk, Bayram Akdemir, Application of Feature Extraction and Classification Methods for Histopathological Image using GLCM, LBP, LBGLCM, GLRLM and SFTA, Procedia Computer Science, Volume 132, 2018, Pages 40-46, https://doi.org/10.1016/j.procs.2018.05.057.

[26] André Ricardo Backes, Jarbas Joaci de Mesquita Sá Junior, LBP maps for improving fractal based texture classification, Neurocomputing, Volume 266, 2017, Pages 1-7, https://doi.org/10.1016/j.neucom.2017.05.020.

[27] Yiming Lei, Ximei Zhao, Guodong Wang, Kexin Yu, Weidong Guo, A novel approach for cirrhosis recognition via improved LBP algorithm and dictionary learning, Biomedical Signal Processing and Control, Volume 38, 2017, Pages 281-292, https://doi.org/10.1016/j.bspc.2017.06.014.

[28] Houssem Lahiani, Mahmoud Neji, Hand gesture recognition method based on HOG-LBP features for mobile devices, Procedia Computer Science, Volume 126, 2018, Pages 254-263, https://doi.org/10.1016/j.procs.2018.07.259.

[29] Neha Sharma, Vibhor Jain, Anju Mishra, An Analysis of Convolutional Neural Networks for Image Classification, Procedia Computer Science, Volume 132, 2018, Pages 377-384, ISSN 1877-0509, https://doi.org/10.1016/j.procs.2018.05.198.

[30] Boukaye Boubacar Traore, Bernard Kamsu-Foguem, Fana Tangara, Deep convolution neural network for image recognition, Ecological Informatics, Volume 48, 2018, Pages 257-268, ISSN 1574-9541, https://doi.org/10.1016/j.ecoinf.2018.10.002.

[31] M.E. Paoletti, J.M. Haut, J. Plaza, A. Plaza, A new deep convolutional neural network for fast hyperspectral image classification, ISPRS Journal of Photogrammetry and Remote Sensing, Volume 145, Part A, 2018, Pages 120-147, https://doi.org/10.1016/j.isprsjprs.2017.11.021.

[32] Shaohua Wan, Yan Liang, Yin Zhang, Deep convolutional neural networks for diabetic retinopathy detection by image classification, Computers & Electrical Engineering, Volume 72, 2018, Pages 274-282, ISSN 0045-7906, https://doi.org/10.1016/j.compeleceng.2018.07.042.

[33] Calvin C. Gotlieb, Herbert E. Kreyszig, Texture descriptors based on co-occurrence matrices, Computer Vision, Graphics, and Image Processing, Volume 51, Issue 1,1990, Pages 70-86, https://doi.org/10.1016/S0734-189X(05)80063-5.

[34] Jirabhorn Kaewchote, Sittichoke Janyong, Wasit Limprasert, Image recognition method using Local Binary Pattern and the Random forest classifier to count post larvae shrimp, Agriculture and Natural Resources, Volume 52, Issue 4, 2018, Pages 371-376, https://doi.org/10.1016/j.anres.2018.10.007.

[35] Topi Mäenpää, Matti Pietikäinen, Classification with color and texture: jointly or separately? Pattern Recognition, Volume 37, Issue 8, 2004, Pages 1629-1640, https://doi.org/10.1016/j.patcog.2003.11.011.

[36] Vijaykumar Digge, Jagannath Desai, Saubhik Das, Expanded Age Indication for Ponseti Method for Correction of Congenital Idiopathic Talipes Equinovarus: A Systematic Review, The Journal of Foot and Ankle Surgery, Volume 57, Issue 1, 2018, Pages 155-158, https://doi.org/10.1053/j.jfas.2017.08.015.

[37] K. R. Singh, and S. Chaudhury, Efficient technique for rice grain classification using back-propagation neural network and wavelet decomposition, IET Computer Vision, 10(8): 780–787(2016).

[38] Longfei Zheng, Yu Wang, D. Jude Hemanth, Arun Kumar Sangiah, and Fuqian Shi, Data augmentation on mice liver cirrhosis microscopic images employing convolutional neural networks and support vector machine, Journal of Ambient Intelligence and Humanized Computing, 2018, https://doi.org/10.1007/s12652-018-0951-8.

[39] L.A. Álvar-Ginés, V.M. Rafael, and E. Kjersti, Noise robust and rotation invariant framework for texture analysis and classification, Applied Mathematics and Computation, 335, 124–132(2018), https://doi.org/10.1016/j.amc.2018.04.018

[40] K. S. Rakesh, K. Ramkumar, S. Srinivasan, V.E. Balas, and F. Shi, Self-calibration of dead reckoning sensor for skid-steer mobile robot localization using neuro-fuzzy systems, in Proc. ITITS 2015, Information Technology and Intelligent Transportation Systems, 2015, IOS Press, Sweden, https://doi.org/10.1007/978-3-319-38771-0_53.

Appendix I. GGCM features definitions

$T1 = \sum_{i=1}^{L-1} \sum_{j=1}^{L_g-1} \dfrac{P_{ij}}{(j+1)^2}$	$T2 = \sum_{i=0}^{L-1} \sum_{j=0}^{L_g-1} j^2 P_{ij}$	$T3 = \sum_{i=0}^{L-1} \left[\sum_{j=0}^{L_g-1} H_{ij}\right]^2 \Big/ \sum_{i=0}^{L-1}\sum_{j=0}^{L_g-1} H_{ij}$
$T4 = \sum_{j=0}^{L_g-1} \left[\sum_{i=0}^{L-1} H_{ij}\right]^2 \Big/ \sum_{i=0}^{L-1}\sum_{j=0}^{L_g-1} H_{ij}$	$T5 = \sum_{i=0}^{L-1} \sum_{j=0}^{L_g-1} P_{ij}^2$	$T6 = \sum_{i=0}^{L-1} i\left[\sum_{j=0}^{L_g-1} P_{ij}\right]$
$T7 = \sum_{j=0}^{L_g-1} j\left[\sum_{i=0}^{L-1} P_{ij}\right]$	$T8 = \sqrt{\sum_{i=0}^{L-1}(j-T6)^2 \sum_{i=0}^{L_g-1} P_{ij}}$	$T9 = \sqrt{\sum_{j=0}^{L_g-1}(j-T7)^2 \sum_{i=0}^{L-1} P_{ij}}$
$T10 = \sum_{i=0}^{L-1} \sum_{j=0}^{L_g-1}(i-T6)(j-T7)P_{ij}$	$T11 = -\sum_{i=0}^{L-1}\left[\sum_{j=0}^{L_g-1}P_{ij}\right]\log_2\left[\sum_{j=0}^{L_g-1}P_{ij}\right]$	$T12 = -\sum_{j=0}^{L_g-1}\left[\sum_{i=0}^{L-1}P_{ij}\right]\log_2\left[\sum_{i=0}^{L-1}P_{ij}\right]$
$T13 = -\sum_{i=0}^{L-1} \sum_{j=0}^{L_g-1} P_{ij}\log_2 P_{ij}$	$T14 = \sum_{i=0}^{L-1} \sum_{j=0}^{L_g-1}(i-j)^2 P_{ij}$	$T15 = \sum_{i=0}^{L-1} \sum_{j=0}^{L_g-1} \dfrac{1}{1+(i-j)^2}P_{ij}$