Design element extraction of plantar pressure imaging employing meta-learning-based graphic convolutional neural networks

Design Element Extraction of Plantar Pressure Imaging Employing Meta-learning-based Graphic Convolutional Neural Networks

Dan Wang [1], Zairan Li [1,*], Nilanjan Dey [2], Rubén González Crespo [3], Fuqian Shi [4,] and R. Simon Sherratt [5]

1. Wenzhou Polytechnic, Wenzhou, PR China
2. Department of Computer Science and Engineering, Techno International New Town, Kolkata, 700156. India
3. Department of Computer Science and Technology, Universidad Internacional de La Rioja, Logroño, Spain
4. Rutgers Cancer Institute of New Jersey, New Brunswick, 08903, USA
5. Department of Biomedical Engineering, the University of Reading, Reading, UK

Abstract: Segmenting plantar pressure images intelligently can provide valuable insight for people with high blood pressure, making bespoke footwear requirements possible and resulting in more comfortable shoe designs. It is, however, difficult to extract design elements from a segmented image dataset. To address this challenge, we propose an ML-GNN model that segments plantar pressure images using metal-earning. The first part of the paper presents a method for extracting image features that reduce the complexity of the ML-GNN algorithm. To create the network structure, we propose optimization meta-based learning. Using a meta-learning-based graphic neural network, we enhance our mask-based CNN prediction model with VGG16 and CNN layers. We pre-processed the plantar pressure dataset using pressure-sensing data acquisition and compared the results. By defining standard image segmentation indices, we demonstrate the high effectiveness of our research. We have developed an ML-GNN model that improves the segmentation accuracy of plantar pressure images and can also be applied to other sensor image datasets. Through our shoe-last customization approach, we enable the shoe industry to manufacture shoes more efficiently, particularly for people with specific healthcare needs who require bespoke shoe designs. Our findings demonstrate the potential of intelligent image segmentation to advance the field of footwear design and improve the lives of people with specific health requirements.

Keywords: Shoe-last design, plantar pressure, meta-learning, graphic convolutional neural networks

1. Introduction

Consumer expectations have closely incorporated the mobile Internet and big data in the past decade. The temple design of products has become synonymous with intelligence. The impact of intelligence on society has been profound as a new driving force for economic and social innovation and development. Interaction design is an important component of intelligent design. Currently, the trend in footwear interactive design is to incorporate human biological information into the interaction design [1,2]. Future footwear manufacturing and business models will be integrated, fully customizable, and disruptive. Bionics, engineering, and artificial intelligence are involved [3]. Intelligent design is also required for rapid industrial production. Intelligent typesetting systems for leather, 3D foot scanning measurements, Pur-micro production lines, and intelligent cutting robots for the shoe industry are key enablers of agile manufacturing. An accurate three-dimensional foot scanner can measure the plantar pressure distribution, which is closely related to footwear comfort. By doing so, the comfort problem can be transformed into a segmentation problem for imaging plantar pressure. Intelligent segmentation of the plantar pressure image can be realized, and the shoe last can be effectively designed for comfort. This research focuses on extracting key design elements [4].

There are two main categories of traditional image enhancement techniques: time- and frequency-domain image enhancement. Image enhancement processes gray pixel values in the image, including gray value transformations, histogram equalization technologies, image smoothing, sharpening techniques, pseudo-color adjustments, and other techniques. In frequency domain image enhancement, the spectrum components of the image are transformed, followed by an inverse Fourier transformation to obtain the result. In general, a certain type of algorithm can only solve one kind of problem that appears in the image. Therefore, in view of the complex problems of infrared images, to improve the image processing effect, multiple algorithms are often used in combination [5]. At present, more algorithms with relatively perfect performance are being proposed. Zotin [6] proposed a fast multiscale retinex (MSR) algorithm to solve the problem of color distortion in the image enhancement process and improved the shortcomings of a time-consuming and slow image enhancement algorithm. However, in processing this algorithm, there are still details that still need to be improved. Hanumantharaju *et al*. [7] proposed a color image enhancement technique based on an improved MSR algorithm where the wavelet energy was used to evaluate the visual quality of the enhanced image. There are several commonly used methods for identifying defects in non-destructive testing images, such as principal component analysis (PCA), pulse phase methods, and signal reconstruction methods. [8].

Traditionally, image recognition involves four steps: image acquisition->preprocessing->feature extraction->recognition. This process involves analysing images to detect patterns in objects and targets with the help of computers. Image recognition algorithms have recently been developed that apply deep learning. Neural network image recognition is based on traditional image recognition algorithms and neural network algorithms. In this case, the neural network refers to an artificial neural network, which means that the neural network is not a real neural network possessed by animals but one that humans have engineered after mimicking animal neural networks. Neural network image recognition technology combines genetic algorithms, back propagation (BP) networks, and neural networks to form classic models, which have many applications. Image recognition systems generally use neural networks to extract the image's features and then map the image's features to neural networks for recognition and classification [9-11]. The feature extraction of plantar pressure images in this study mainly applies the method of deep neural networks.

2. Literature Review

In the first paper on graph neural networks, Gori *et al.* [12] introduced the concept, and Scarselli *et al.* [13] offered further clarification. These early studies propagated neighbor information iteratively through RNNs to learn the representation of target nodes until a stable fixed point is reached. Recent methods for data convolution have been redefined by the success of convolutional networks in application domains such as computer vision. Bronstein *et al.* [14] described deep learning methods in non-Euclidean domains, including graphs and manifolds. Although this was the first review of graph convolutional networks, their study did not discuss several important spatial-based approaches.

Graph neural networks (GNNs) and convolutional neural networks (CNNs) can be used together for image segmentation tasks. In this context, GNNs can be used to model the relationships between pixels in an image as a graph, where nodes represent individual pixels and edges represent the relationships between them. The GNN can then learn representations of these relationships and propagate information through the graph to refine these representations. This can be particularly useful for capturing long-range dependencies in the image, which may be difficult for CNNs to capture. Once the GNN has learned these representations, they can be fed into a CNN to perform the actual segmentation task. The CNN can use these representations to guide its feature extraction and produce more accurate segmentation results [15]. Overall, the combination of GNNs and CNNs allows for a more robust and accurate approach to image segmentation, particularly in cases where long-range dependencies are important. [16]

Metalearning (ML) consists of completing the general process of machine learning and designing network structures, such as CNNs and recurrent neural networks (RNNs). ; selecting a distribution to initialize parameters (in practice, choosing different network structures or parameters is equivalent to defining different functions); inputting training data, calculating the loss according to the loss function; gradient descent, updating step by step; and obtaining the final function. However, the model needs to be retrained when the scene is changed [17]. In addition to the network structure, the parameters are artificially designed and called "hyperparameters". In ML, it is expected that these configurations, such as network structure, parameter initialization, optimizer, etc., are determined by the machine itself so that the network performance improves [18-20]. Graph convolutional networks (GCNs) form the basis of many complex graph neural network models. There are two categories of GCN methods: spectrally based and spatially based. In graph signal processing, spectral-based methods define graph convolution as a way of removing noise from graph signals. Convolution of graphs is commonly described as an aggregation of neighboring feature information. By alternating GCN layers with graph pooling modules [21-23], fine-graining can be achieved. Deep neural architecture is the key to deep learning's success. ResNet, for example, has 152 layers for image classification. As the number of layers increases in graph networks, however, model performance drops dramatically. Due to graph convolution, adjacent node representations are essentially pushed closer together [24, 25].

Metalearning (ML) with graph neural networks (GNNs) for image segmentation is novel because it can rapidly adapt to new datasets and improve segmentation accuracy with limited data. GNNs are traditionally used to segment images by modelling pixel relationships as graphs and learning representations of those relationships. In addition, this approach can be computationally expensive and requires a large amount of labelled data for training. Our approach can improve segmentation accuracy with fewer labelled examples by incorporating meta-learning into the GNN model. A meta-learning framework enables a GNN to learn how to learn, i.e., how to quickly adapt to new data with limited labelled examples. As part of our approach, we employ an image feature extraction method to reduce the complexity of the ML-GNN algorithm and a meta-based optimization approach to create the network structure. The ML-GNN's efficiency and accuracy for image segmentation are further improved by adding these components. ML-GNN provides an efficient and accurate approach to image segmentation by adapting to new datasets quickly, improving segmentation accuracy with limited data, and providing a faster and more accurate approach than existing approaches. This work makes the following novel contributions to research:

(1) Our model leverages meta-learning techniques to adapt and generalize well to different plantar pressure image datasets. This enables the model to learn from multiple tasks or datasets, acquiring knowledge that can be applied to improve

segmentation accuracy on new, unseen datasets. The application of meta-learning in image segmentation is a novel approach that has the potential to enhance the performance and generalizability of segmentation models in various domains.

(2) We propose an optimization meta-based learning method to create the network structure of our ML-GNN model. This approach optimizes the structure of the GNN by leveraging meta-learning techniques, leading to improved segmentation accuracy. The integration of metalearning and GNNs for image segmentation is a novel combination that has not been extensively explored in the literature.

(3) The primary application of our model is in the field of footwear design. By accurately segmenting plantar pressure images and extracting design elements, our model enables bespoke footwear requirements for individuals with high blood pressure or specific healthcare needs. This aspect of our research has implications for the efficient manufacturing of shoes and the improvement of comfort and fit, particularly for individuals with specific health requirements. While the use of VGG16 is a standard practice in image processing, the application of our ML-GNN model to address footwear design challenges is a novel contribution.

The remaining sections are organized as follows. The modelling of a metalearning-based graph neural network for segmenting plantar images is presented in Section II. In section III, the experimental design is described, as well as the data acquisition and processing followed by the results. Section IV presents a discussion, and our conclusions are presented in Section V.

3 Modelling

3.1. Metalearning for Neural Network Parameter Training

A meta-learning approach to neural network parameter training can be used to train the optimizer itself, making it more adaptable to new tasks. Metalearning involves training an optimizer to learn how to learn by using a small set of tasks. Meta-training involves learning how to update the network parameters to achieve good performance on the training tasks. Using small adjustments to the network parameters, the optimizer can quickly adapt to new tasks once it has been meta-trained. With fewer examples, the network can learn faster. Using a meta-learning algorithm such as MAML (Model-Agnostic Meta-Learning) or Reptile to train the optimizer is a common way to meta-learn neural network parameters. Based on the performance of the network on a small set of tasks, these algorithms update the optimizer's parameters. Once trained, the optimizer can quickly adapt to new tasks by fine-tuning the network parameters or generating new parameters based on the optimizer's findings. Metalearning provides a powerful approach to quickly adapt to new tasks and improve the efficiency of learning neural networks [26].

The purpose of meta-learning is to automatically train the function in the training task and then use this prior knowledge to train the parameters in the model under a specific task in the test task, as shown in the following dependencies:

$$F_{\varphi,\theta} \mapsto \left(F_{\varphi',\theta} \Leftrightarrow f_{\theta}\right) \mapsto \left(F_{\varphi',\theta} \Leftrightarrow f_{\theta'}\right) \tag{1}$$

When training a neural network, the specific general steps are preprocessing the dataset, choosing the network structure, setting hyperparameters, initializing the parameters, choosing the optimizer, defining the loss function, and gradient descent updating the parameters. The path of machine learning can be summarized as follows: preprocessing dataset D→select network structure N→set hyperparameter γ → initialization parameter θ_0 →select optimizer O→define loss function L→gradient descent update parameter θ_0. The path of meta-learning can be summarized as follows: learning to preprocess dataset D→ learning to select network structure N→learning to set hyperparameter γ → learning initialization parameter θ_0 → learning to select optimizer O→ learning definition loss function L→ gradient descent update parameter θ_0. Metalearning learns all the parametric variables that need to be set and defined by humans. Here, the parameter variable belongs to the set, therefore:

$$\varphi \in \Phi = \left\{D, N, \gamma, \theta_0, O, L\right\} \tag{2}$$

As part of conventional deep learning, we apply gradient descent to training examples (such as classifying pictures of animals into one of five species) to learn optimal parameters. A meta-learning algorithm uses the task itself as a training example: to learn optimal parameters for a particular problem type, we apply learning Algorithm 1 as follows. [27].

Algorithm 1: basic meta-learning model

REQUIRED: $P(T)$-distribution over task; α , β - step size.

OUTPUT: θ

θ=0

FOR *i* **IN** batch

sample_batch<-sampling $\left(P\left(T(i)\right)\right)$

FOR T(j) **IN** samples(i)

Evaluate $\nabla_{\theta} L_{T(j)}\left(f(\theta)\right)$ subject to *K* samples (i)

Compute $\theta_j^{'} = \theta - \alpha \nabla_\theta L_{T(j)}\left(f(\theta)\right)$ for sample (i)

Sampling $D_j^{'} = \left\{x^j, y^j\right\}$ from $T(j)$ for the meta of sample (i)

ENDFOR

UPDTAE $\theta = \theta - \beta \nabla_{T(i)\sim L_{T(i)}\left(f\left(\theta_i^{'}\right)\right)} L_{T(i)}\left(f\left(\theta_i^{'}\right)\right)$

ENDFOR

3.2 Graphic Neural Networks for Image Feature Extraction

For a given graph, each node has a certain feature; suppose that X_v represents the feature of node v, as do the edges that connect the nodes. Let $x(u,v)$ be the edge that represents the edge between node v and node u; the learning goal of graphic neural networks (GNNs) is to obtain the hidden state h_v (state embedding) of each node's graph perception; in other words, for each node, its hidden state contains information from its neighbours. Therefore, it is necessary to determine how each node on the graph can be aware of its neighbors. A GNN is implemented by iteratively updating all nodes' hidden states. The hidden state of node v is updated as follows at time $t+1$:

$$h_{v+1} = f\left(X_v, X_{co}[v], h_{nte}[v], X_{nte}[v]\right) \tag{3}$$

where $f(\cdot)$ is the state update function of the hidden state, which in this work is also called the local transaction function. $X_{co}[v]$ in the formula refers to the feature of the edge adjacent to node v, $X_{ne}[v]$ refers to the feature of the neighbor node of node v, and $h_{nte}[v]$ refers to the neighbor node at time *t* hidden state. By combining all $f(\cdot)$, an artificial neural network (ANN) was suggested, and it is worth mentioning that it seems that the input of f is a variable-length parameter. This new formula updates the hidden state of the central node based on the hidden state of the neighbor nodes at the current moment until very little variation exists in the hidden state of each node and the information flow of the whole graph is stable. The neighbors of each node now know what each knows. A second function g is needed to indicate how to adapt to upcoming tasks in addition to the state update formula, which explains how to find hidden states. In addition to expressing local output functions g, neural networks can also represent globally shared functions g. The process can be described as follows using neural networks:

Connect the two moments based on the graph's connection. The state of node 1 accepts the hidden state of node 3 at time T_1 since node 1 is adjacent to node 3. After Tn iterations, each node's hidden state converges, and it is followed by a g to obtain its output o. The convergence time may differ for different graphs since convergence is determined by whether the difference between the p-norms of the two moments is less than a certain threshold ε, $\|h_{t+1}\|^2 - \|h_t\|^2 < \varepsilon$. The input for the gradient operation is the number of timesteps (Tn), and the plantar pressure image features (Xv) we extracted in previous steps.

Step 1: For N training tasks regarding the support set (SS) and query set (QS), a test task is applied for evaluation of the meta-learning parameters.

Step 2: Define the network structure; initialize a meta network θ_0.

Step 3: Pretrain iteratively:

1) Assign the initial meta-parameter θ^0 to the m^{th} task to obtain $\hat{\theta}^m$
2) Optimize $\hat{\theta}^m$ using the SS of the m^{th} task based on the learning rate of α_m
3) Compute the loss of the m^{th} task $l^m\left(\hat{\theta}^m\right)$ using QS
4) Update the gradient ϕ^0 to ϕ^1 using the learning rate α_{meta}
5) Sample a task n, set $\hat{\theta}^n = \phi^1$
6) Optimize the learning rate of α_n and update $\hat{\theta}^n$
7) Compute loss $l^n\left(\hat{\theta}^n\right)$ using QS
8) Update ϕ^1 to ϕ^2
9) Go to STEP 3.1 until all tasks are finished

Step 4: Fine-tune the meta network.

Step 5: Evaluate QS in meta-learning.

3.3 Metalearning-Based Graphic Neural Network

3.3.1 Optimization Meta-Based Learning

In few-shot learning, meta-learning is applied as supervised learning. Fig. 1 illustrates three types of models used in few-shot learning, namely, the model-based method, the metric-based method, and the optimization-based method. The model-based method is used to determine how the input data (X) can be translated into predicted values (P) and how the parameter values can be quickly adapted based on a small number of samples. This method typically involves training a meta-model that can generate task-specific models with different parameter values to adapt to new tasks during meta-training. During meta-testing, the task-specific models are quickly adapted to new tasks using a small amount of task-specific data. The metric-based method measures the distance between inputs in the batch set and the inputs in the support set, which contains a few labelled examples per class. The nearest neighbor approach is used to classify new instances by comparing their distances to the support set. This method typically involves learning a distance metric or embedding space that can capture the similarity between samples and using this metric to compute distances and make predictions. The optimization-based method adjusts the ordinary gradient descent method to fit small samples in few-shot scenarios. This approach usually involves designing specialized optimization algorithms that can effectively update model parameters with limited data. This can include techniques such as using higher-order optimization methods, incorporating memory-augmented neural networks, or designing adaptive learning rate schedules. The choice of which method to use in few-shot learning depends on the specific problem, data, and constraints of the task at hand. Each method has its strengths and weaknesses, and researchers often explore combinations of these approaches to achieve the best performance in different few-shot learning scenarios [28].

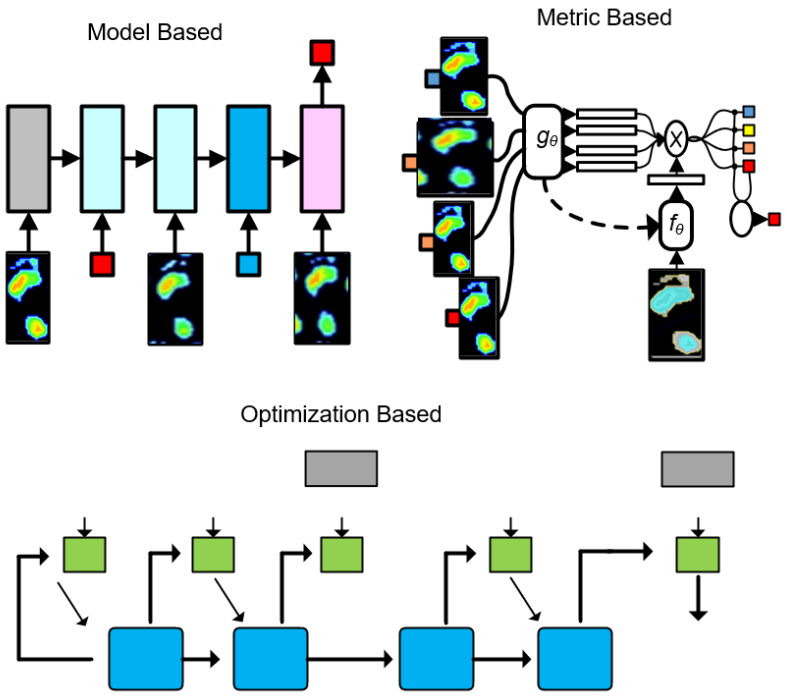


Fig. 1. Meta-based learning models in the context of basic metric-based and optimization-based approaches in few-shot learning.

3.3.2 Semantic Segmentation Model

Semantic segmentation is a process that involves assigning labels to each point in an image based on the category to which it belongs. This is achieved through pixel-level classification, where the input and output of the system are represented as one-hot vectors for each pixel in the image. Convolutional neural networks (CNNs) are commonly used for semantic segmentation, with full convolutional networks (FCNs) being the first to use an encoder-decoder structure and information fusion through concatenation [29, 30]. U-Net and SegNet are two popular CNN architectures that use the encoder-decoder structure and information fusion principles. To overcome information loss caused by multiple pooling layers, SegNet uses pooling with coordinates to preserve location information. DeepLab V1 uses atrous convolution to control the size of the receptive field and adds a conditional random field (CRF) to use correlation information between pixels. PSPnet fuses feature maps of shallow and deep layers to combine semantic features with different receptive fields [31-33]. DeepLab V2 introduces atrous spatial pyramid pooling (ASPP), which selects atrous convolutions with different rates to process feature maps and integrates information from different levels. DeepLab V3 improves the ASPP module by adding batch normalization and global average pooling, which emphasizes global features and overcomes weight degradation issues. DeepLab V3+ uses DeepLab V3 as an encoder and adds Xception to reduce the number of parameters and improve the running speed [34,35].

(1) Basic Meta with Few Shots

The dual-branch network is used for few-shot segmentation, the conditional branch uses VGG to extract features and generates weights (w, b), and the segmentation branch uses the FCN-32s structure to extract features from the query image, which is multiplied by the parameters obtained from the conditional branch, as shown in Fig. 2. Then, the segmentation result is obtained by the σ function. Obtain a segmentation map, upsample to the image size, and use a certain threshold to generate a segmented binary map. When outputting, to make the parameter quantity correspond to the number of channels of the feature map of the segmentation branch, the weight hashing strategy is used to map the output 1000-dimensional vector to 4097-dimensional (w:4096, b:1). This mapping mechanism is modelled as a fully connected layer with fixed weight parameters [36].

The segmentation branch is used to extract features from the query image, concise the result with the embedding obtained from the conditioning branch, and then perform pixel-level segmentation. Fig. 3 shows the query images from the positive support set using VGG16 and CNN layers. Then, in the task flow, the prototype of each category is continuously updated, as shown in Fig. 4.

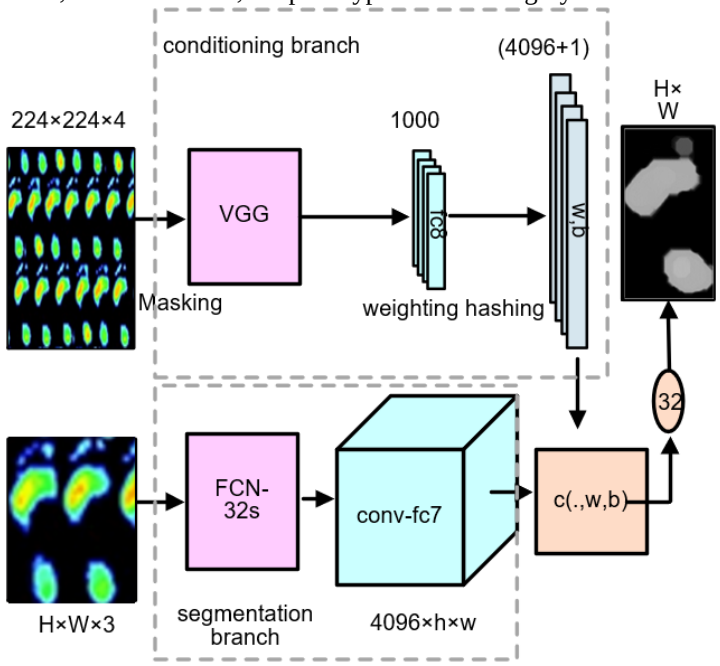


Fig. 2. Segmentation and conditioning branches using VGG and FCN-32s.

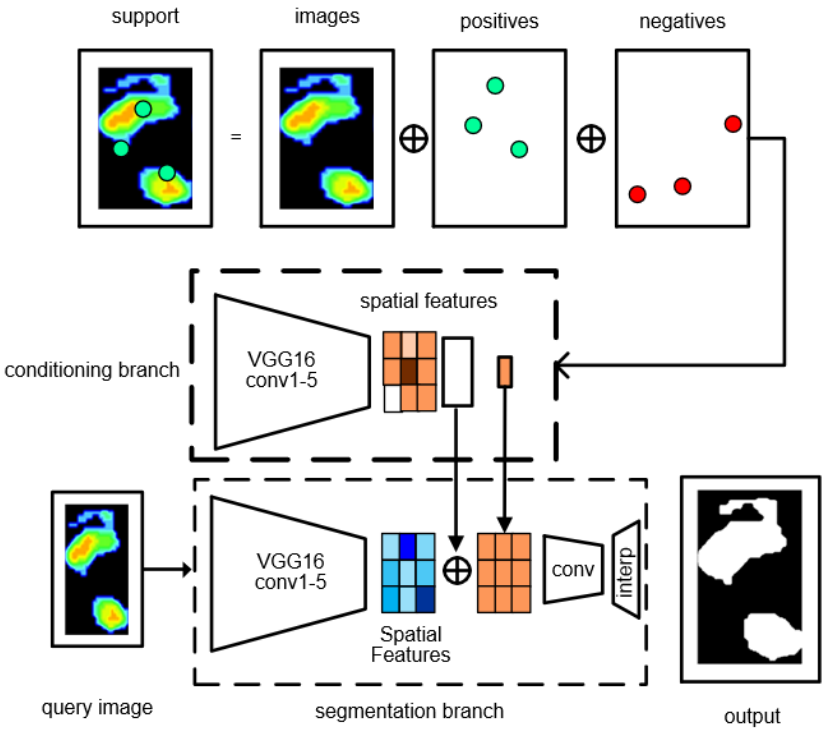


Fig. 3. Query images from the positive support set using VGG16 and CNN layers.

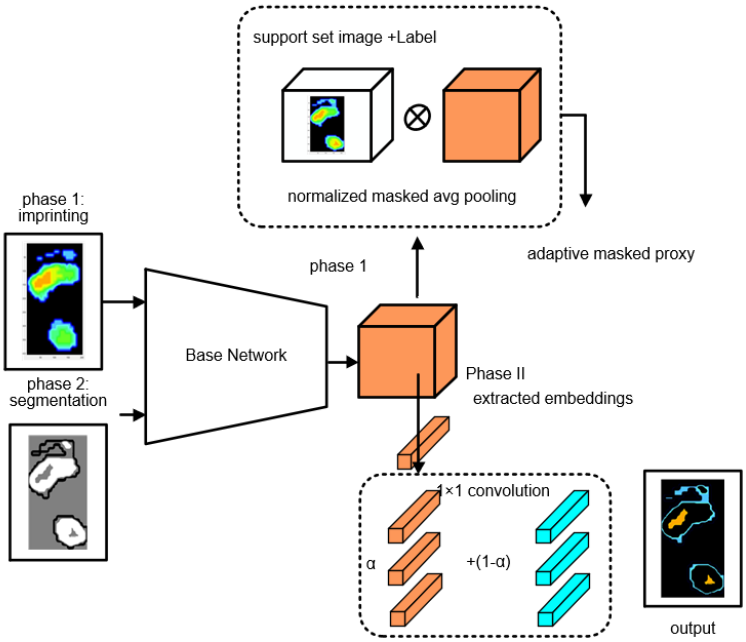


Fig. 4. Two phases of the base network combined with the convolutional network with SS.

(2) One-Shot Semantic Segmentation

The network consists of VGG-16 and two branches: guidance and segmentation. It should be noted that the two branches share three convolution blocks. Personal understanding is that if the two branches are completely independent, then the guidance generated by the Guidance Branch will remain unchanged; this loses its meaning, and the author performed relevant analysis and experiments in the final ablation experiment of the paper. After interacting with the segmentation branch, the guided feature map can be optimized to match the corresponding real mask while optimizing the segmentation loss. The process of the training tasks is shown in Fig. 5.

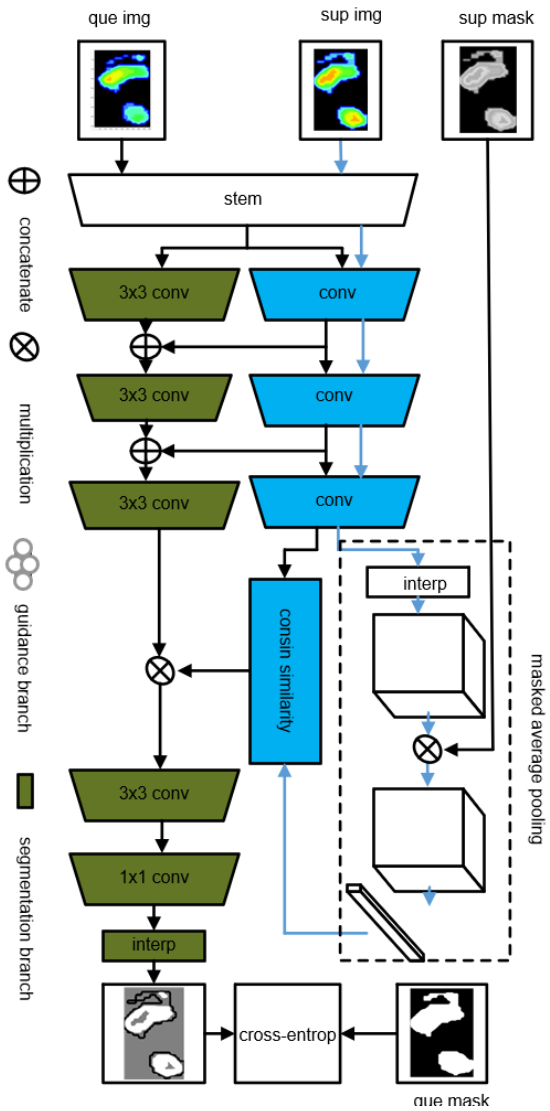


Fig. 5. Multiple convolutional layers with guidance and segmentations based on cross-entropy.

The network is loaded with weights pretrained on the ILSVRC dataset during training. Leveraging metric learning on prototypes, no parameters are proposed in the regularization of prototype alignment, and we make full use of the knowledge of support. It may directly be used for few samples with weak annotations, as shown in Fig. 6.

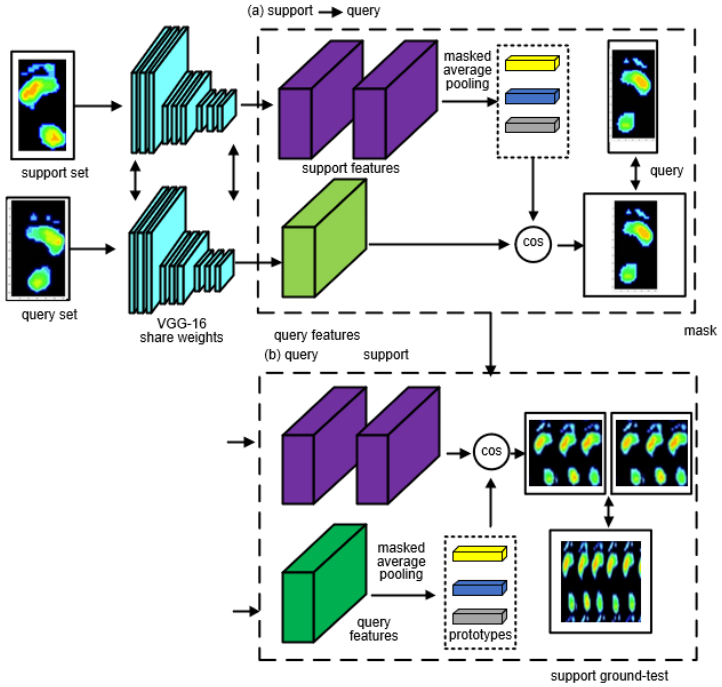


Fig. 6. SS (support set) and QS (query set) for VGG-16 using MAP (masked average pooling).

The same backbone is used to extract the depth features of support and query, and then masked average pooling is used to embed different foreground objects and backgrounds into different prototypes from the features of support. Each prototype represents the corresponding category so that each of the query images has a pixel labelled by referencing the class-specific prototype closest to its embedding expression after obtaining the predicted mask of the query. During training, after getting the mask, use the query feature and mask just extracted as the new "support set", use the previous support set as the new "query set", and then use the "support set" to do the "query set" A wave of predictions, and then a loss. The prototype is compact and robust to represent each semantic category; the mask marked block is parameter-free metric learning, and segmentation is performed by pixel-by-pixel matching with the embedding space. Fig. 7. The support set and query set based on feature extraction for mask operation.

To perform the prototype alignment regularization, it is required to build a new support with the query and its mask and then use this to predict the split of the original support set. Experiments have shown that it can encourage query prototypes to align their support prototypes only during training. The feature boosting is found by:

$$f_s^{n+1} = f_s^n + v\frac{\partial L(\hat{m}_s^n, m_s)}{\partial f_s^n} \tag{4}$$

Fig. 8 shows the support mask-based CNN prediction model using boosting proposed in Eqn. (4).

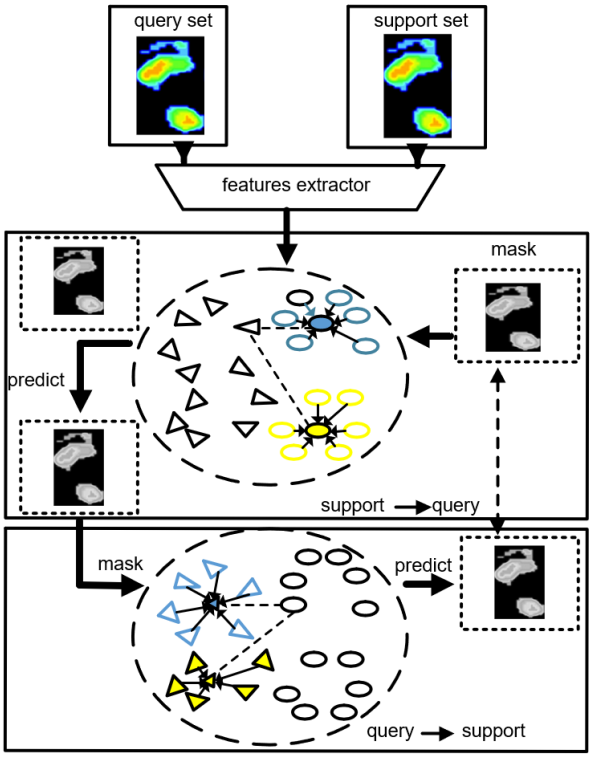


Fig. 7. Pipeline for plantar pressure images by using SS (support set) and QS (query set) based on feature extraction for mask operation.

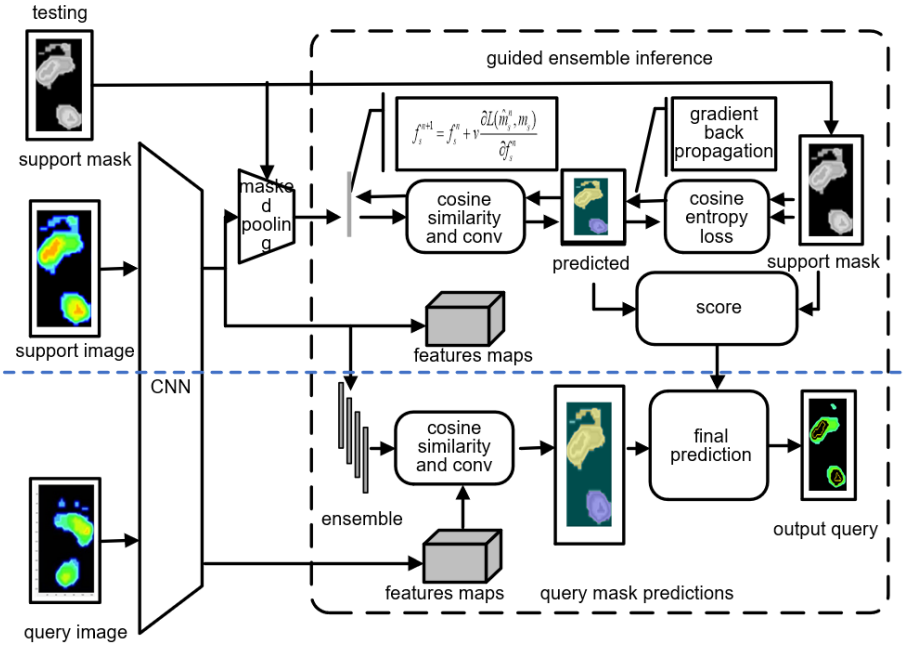


Fig. 8. The support mask-based CNN prediction model for plantar pressure image segmentation showed better performance by using query and support in the model

4. Experimental Design and Results

4.1. Experimental Design and Data Acquisition

A Foot-Scan plantar pressure distribution test system (RSscan, Belgium) was used in this study that adopts micro pressure sensor technology. From the perspective of biomechanics, the gait characteristics of people during walking or exercise are obtained through dense, high-frequency pressure distribution plates. Using dynamic pressure distribution data of the sole and the heel of the shoe, one can evaluate the biomechanical properties of shoes, such as cushioning, support, stability, etc. [37].

4.2 Experimental Design of Pressure Sensing Data Acquisition

4.2.1 Subjects and Experimental Preparation

One hundred healthy participants without flat feet (50 males, 50 females) were tested. Static and dynamic test experiments were carried out through a wide and straight-through channel with a length of 6 meters (auxiliary 2 meters). Due to the different walking posture dynamics of individuals, there is a great disturbance to the stability of plantar pressure in the gait test experiment. Therefore, before the experiment, it is necessary to introduce the experimental equipment and precautions to the subjects. A sample picture of the basic test method and posture introduction was posted on the wall on the horizontal side of the experiment, and simple gait training was performed for all subjects. It was required to keep the eyes horizontally forward and keep the walking posture when walking; this measure ensures the objectivity and validity of the experimental data (data available at GitHub repository: https://github.com/zairanli/plantar).

4.2.2 Test Requirements

The "intermediate step" method of selecting the upper board method involves collecting data 4-6 times. It is necessary to collect at least eight data collections for "one-step loading" and at least five data collections for "two-step loading" to obtain reliable "peak pressure" and "pressure–time integration". To obtain a more reliable pressure peak and pressure time, the integral value requires at least 6 data acquisitions. "Three-step loading" requires at least 6 data collections. When conducting a static balance test, control the time when testing the four states (double-open, double-closed, single-open, and single-closed). Usually, stand on two feet for 20 seconds and on one foot for 10 seconds. The valid criteria for the test are as follows: the footprints displayed on the pressure plate are complete; during the test, the subject looks forward and goes up the plate naturally; and there is no obvious gait change on the plate.

4.2.3 Plantar Pressure Image Acquisition Results

Most studies are usually carried out using the index system presented in Table 1, and the methods used are based on statistics.

Table 1. Plantar Pressure Data Collection Index and its Definition

Items	Explanation
Standard warp peak pressure	The sum of the pressure values at the corresponding time point is divided by the area of the landing site of the corresponding area at the current time point
Standard peak force	Refers to the sum of the force sensor values at the current time point in the zone
time to peak pressure	The percentage position of the time points of the normalized peak pressure in the support phase
time to peak force	The percentage position of the time points of the normalized peak force in the support phase
landing time	Percentage of each zone's long landing time in the support phase
The most initial landing time	The time point when each feature partition first hits the ground is in the support stage.
Impulse	Normalized pressure or integral of pressure over time

The Foot-Scan imaging system can generate left and right foot pressure images at different times. To achieve successful image fusion, image preprocessing is a crucial step. Different fusion algorithms require appropriate preprocessing techniques. To accomplish this, background removal, dimensionality reduction, filtering, segmentation, and other tasks are necessary. The foot scan experiment involved 100 volunteers aged 18 or over. To participate in the plantar pressure imaging experiment, potential volunteers needed to have normal neurological functions without any related neurological diseases, no walking instability, abnormal gait, intermittent claudication, or blurred vision. Additionally, they had to have normal muscle strength and tendon reflexes, no severe foot pain conditions, and no foot ulcers. The volunteers completed relevant forms before providing medical data. They were asked to remove their socks and report their shoe-wearing habits, age, gender, height, weight, and blood pressure. During the test collection, volunteers walked at a normal speed for 10 repetitions. The plantar pressure imaging system divides the plantar into 10 anatomically based divisions and provides measurement data (in discrete values), including medial, lateral, midfoot, five metatarsal bones, foot size big toe, and the other four toes. Partial plantar pressure imaging results are presented in Fig. 9, where the colors representing different pressures are automatically calculated and converted by Foot-Scan.

Fig. 10 presents a single image of plantar pressure (left foot), in which the line above the image is the central pressure value automatically calculated by the system. Fig. 10(a) presents the distribution of the plantar pressure when the experiment is carried

out to the maximum contact surface of the whole foot; Fig. 10(b) presents the distribution of the plantar pressure when the subject's forefoot touches the ground.

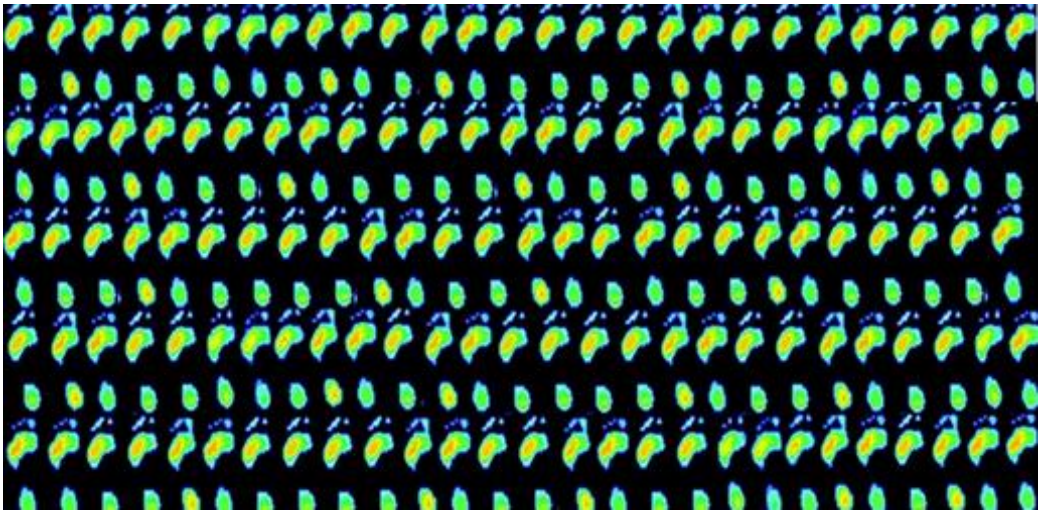
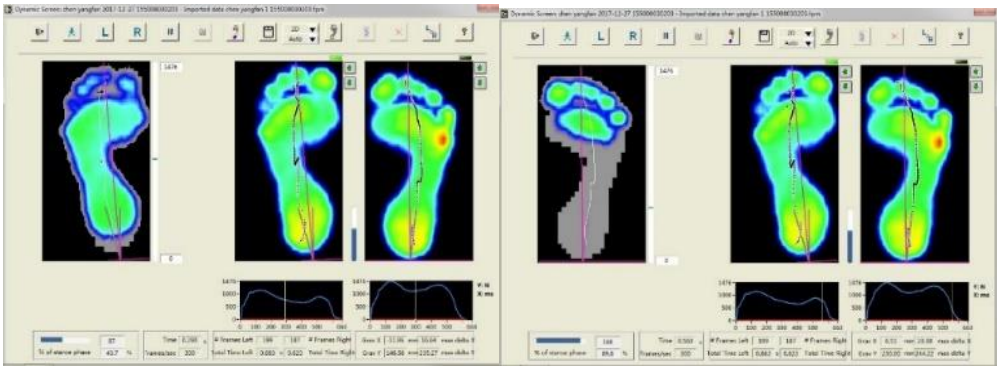


Fig. 9. Partial plantar imaging data collected by plantar pressure experiment.

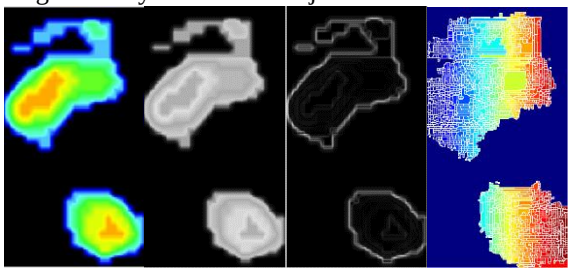


(a) (b)

Fig. 10. The dynamic change process of plantar pressure; (a) maximum state of full foot contact surface; (b) the state when the front foot touches the ground.

4.2.4 Data Preprocessing and Feature Extraction

Traditional watershed-based and color-based segmentation methods cannot satisfy the accurate segmentation of plantar pressure images, as shown in Fig. 11. Image segmentation yields an image that is made up of an array of segments or a series of contours (see edge detection). An area comprises pixels that are similar in some characteristic or computed property, such as color, intensity, or texture. The same characteristics are significantly different in adjacent areas.



(1) (2) (3) (4)

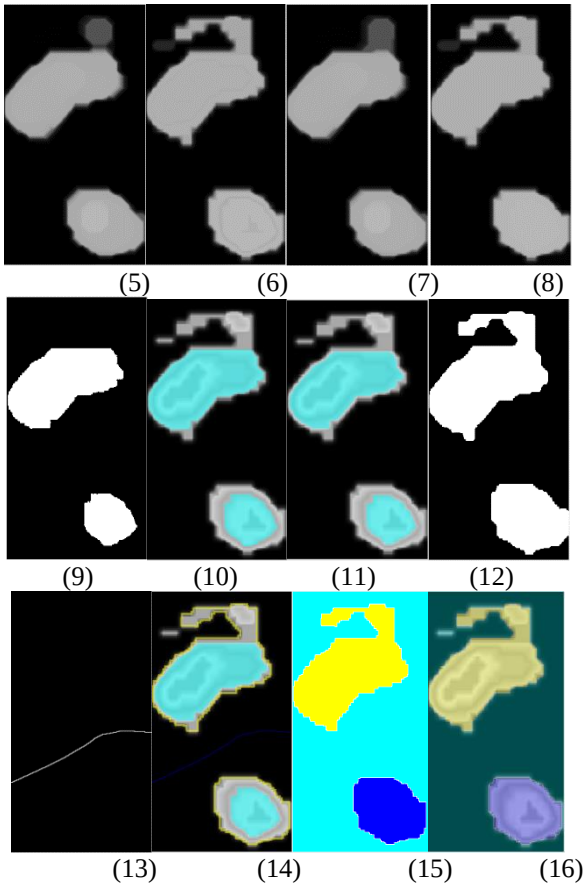


Fig. 11. Examples of watershed segmentation results.

Fig. 12 presents the color image segmentation result using the proposed ML-GNN.

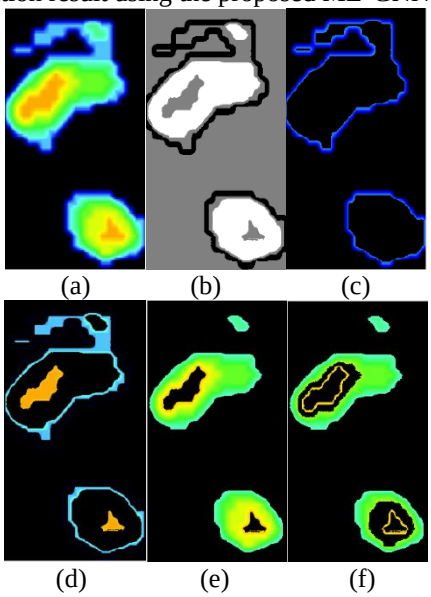


Fig. 12. Color-based image segmentation process; (a) original image, (b) grayscale k-clustering, (c) class-1, (d) class-2, (e) blue kernel segmentation, (f) ML-GNN result.

4.3 Segmentation Results

The plantar pressure image set is preprocessed to reduce dimensionality and detect intermediate states, as shown in Fig. 13 and preprocessed in Fig. 14. The results of the texture filtering segmentation, color K-means [38], ordinary watershed, and gradient watershed segmentation of the plantar pressure imaging set are shown in Fig. 15. As mentioned above, these methods are all typical segmentation methods used currently, and the evaluation indicators are used to compare them with the proposed morphological segmentation method.

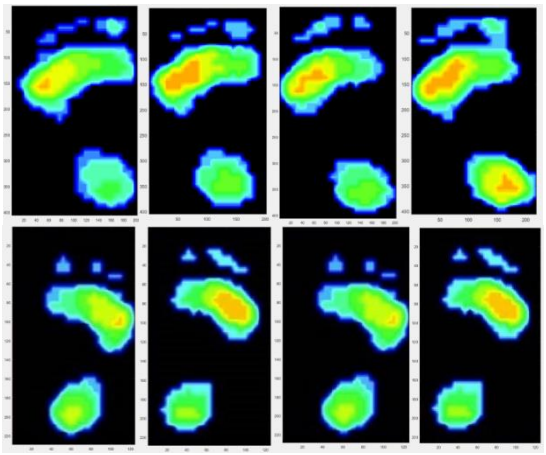


Fig. 13. Original plantar pressure imaging.

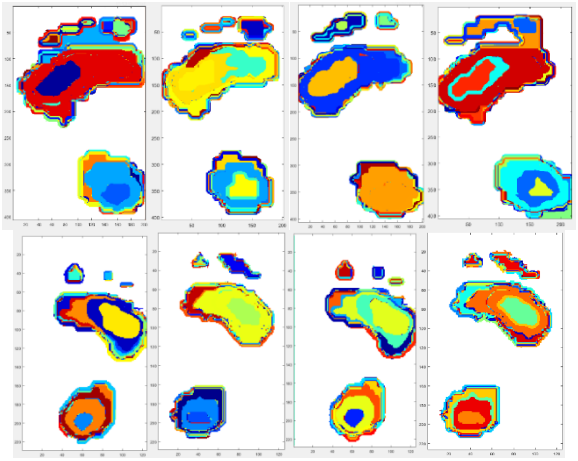
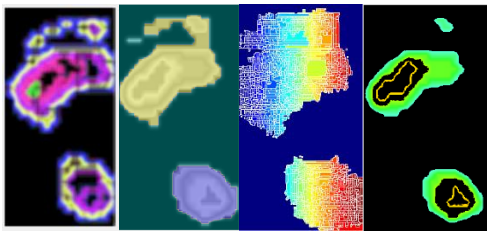


Fig. 14. Calculation results of eight pictures of left and right feet using meta-learning with CNN.



(a) (b) (c) (d)

Fig. 15. Various segmentation methods for comparison analysis: (a) texture-filtered segmentation, (b) normal watershed, (c) gradient watershed, and (d) meta-learning with GNN.

4.4. Discussion

Several evaluation metrics and criteria are commonly used to evaluate the accuracy and reliability of image segmentation algorithms. Pixel-by-pixel labelling includes two common accuracy criteria: pixel accuracy and intersection over union; commonly used accuracy criteria include:

1) Pixel accuracy (PA): This is a simple but useful metric that measures the percentage of pixels that were marked correctly to the total number of pixels. As an alternative metric for evaluating image segmentation, one can report the percentage of pixels correctly classified in an image. Typically, each pixel class's accuracy is reported separately, as is the overall accuracy.
2) The mean pixel accuracy (MPA) is the average of the PAs. Taking the average of all the classes, calculate the percentage of pixels in each class that are correctly classified. The sensitivity curve (SC) is sometimes referred to as this curve.
3) The receiver operating characteristic (ROC) curve is a two-dimensional coordinate graph in which the horizontal axis represents the false positive probability and the vertical axis represents the true positive probability. Based on different judgment criteria, a curve is drawn under stimulation. As with traditional evaluation methods, the ROC curve's evaluation method considers actual conditions, allowing intermediate states to appear, and dividing test results into categories, such

as normal, roughly normal, suspicious, roughly abnormal, and five degrees of anomaly. Thus, ROC curves appear to be more widely applicable. If continuous values are not available rather than binary images, ROC curves cannot be used to evaluate image segmentation algorithms. An image segmentation algorithm can be evaluated multidimensionally using the average precision rate, recall rate, and F1 index. From an already segmented binary image, several segmented binary images may be created. If the segmentation result is not large enough, it is difficult to obtain the ROC curve.

4) Dice coincidence rate: It is a set similarity measure function, which is the same as the similarity index. It represents the segmentation result of the calibrated ground truth (GT), and V_{seg} represents the predicted segmentation result:

$$DICE = \frac{2 \times \left(sim\left(V_{seg}, V_{gt} \right) \right)}{V_{seg} + V_{gt}} \tag{5}$$

where $sim\left(V_{seg}, V_{gt} \right)$ represents a similarity defined in the Euclidean distance space. It is a common practice to calculate the proportion of correctly classified pixels in each class and then to find the average center of all classes.

5) Mean intersection over union (MIoU): Image segmentation is a measure that is based on calculating the intersections and unions of two sets of data. This is the prediction segmentation (PS) of the true value and the predicted value in the image segmentation problem. By dividing the true number by the sum of true, false negatives, and false positives (union). IoU is calculated on each class and averaged afterwards.

6) Frequency weighted intersection over union (FWIoU) is an improvement of MIoU. This method sets weights for each class according to the frequency of occurrence. Due to its simple calculation and strong representation, MIoU is currently the most popular of all the metrics mentioned above. In the process of segmenting an image, it is always necessary to evaluate the results quantitatively. To assess the algorithm's segmentation efficiency, segmentation accuracy, oversegmentation rate, undersegmentation rate, etc., are currently available as metrics for segmentation image calculation based on calibrated ground-truth images. As a reference image for comparing the resulting image, the GT image can be used, that is, the image containing the theoretical segmentation result. For example, GT images are usually drawn manually by experts, and their theoretical values can be derived. This operation can be performed by multiple experts, giving the result of multiple GT segmentations and then averaging this value for each of the evaluation parameters. Segmentation accuracy (SA) is the percentage of the accurately segmented area to the real area in the GT image as follows:

$$SA = \left(1 - \frac{\left| R_s - T_s \right|}{R_s} \right) \times 100\% \tag{6}$$

where R_s represents the reference area of the segmented image roughly sketched by domain experts by hand; T_s represents the real area of the image segmented by the algorithm; and $\left| R_s - T_s \right|$ represents the number of incorrectly segmented pixels.

7) The oversegmentation rate is the ratio of pixels that are segmented outside the reference area of the GT image. The specific calculation formula is as follows:

$$OR = \frac{O_s}{R_s + O_s} \tag{7}$$

where O_s represents the number of pixels that should not be included in the segmentation result but appear in the segmentation result. In other words, the pixels in O_s appear in the actual segmented image but not in the theoretical segmented image R_s.

8) In a GT image, undersegmentation is expressed as the ratio of missing pixels compared to the reference area. To calculate undersegmentation, use the formula below:

$$UR = \frac{U_s}{R_s + O_s} \tag{8}$$

where U_s represents the number of pixels that should be included in the segmentation result but do not appear in the segmentation result. That is, the pixels in U_s appear in the theoretical segmented image but not in the actual segmented image.

The calculation of each index is based on the mean value preservation form method, that is, the data of other indicators are obtained by dividing each index by the value of the mean value preservation form method. Table 2 shows that the proposed ML-GNN is superior to the compared methods.

Table 2. Evaluation Index of Segmentation Effect (Standardized To [0,1])

Index	**Methods**				
	Texture Filtering	**Color K-means**	**Ordinary Watershed**	**Gradient Watershed**	**ML-GNN**

PA	0.877	0.892	0.702	0.634	1
MPA	0.765	0.812	0.792	0.629	1
MIoU	1.123*	0.834	0.871	0.842	1
FWIoU	0.812	0.812	0.932	0.812	1
GT-SA	0.887	0.762	0.829	0.709	1
OR	1.123	0.912*	1.422	1.432	1
UR	0.865	1.120*	0.912	0.567	1
DICE	0.777	0.883	0.676	0.832	1

(*) Indicates that the method index is inferior to the comparison method; the mean value preservation form (this method) is used as the standard reference value 1.

There are some limitations for this research, while our study mainly focuses on the application of our proposed method in a controlled setting with healthy participants, we recognize the importance of investigating its applicability in populations affected by different conditions such as obesity, diabetes, Parkinson's disease, and others [39]. In future research, we plan to explore these connections and assess the adaptability and effectiveness of our approach in diverse clinical contexts, including those involving individuals with specific medical conditions. And also, individuals with large feet or unique foot shapes, which may not be adequately represented in our training data. To address this issue, we may employ techniques such as data augmentation and regularization to enhance the robustness of our model against variations in input data in future. On other hand, algorithm 1 with Eqn(1), and Eqn (3) for feature extraction need to be optimized in adapt to VGG and CNN, that we used for the middle-layers in the proposed model [40].

5. Conclusions

In the field of design methodology, optimal design and intelligent design are hot topics. Computers, biomechanics, and artificial intelligence are also involved in comfort research. According to preliminary research, the design of the last body shape, the structure of the upper surface, the bottom process, the choice of materials, and the microenvironment of the shoe cavity influence the wearing comfort of the footwear. The design of the last surface is the most influential factor. Controlling the generation of the last surface based on the pressure dataset of dynamic foot images and fusing the results can improve the wearing comfort of footwear products. Research directions in footwear comfort design include dynamic image preprocessing, pressure image feature region selection, image intelligent recognition, and dataset fusion. Shoes play an important role in the design of footwear products. The shape of the last body and the design of the sole greatly affect the comfort performance of a shoe. Measurement methods and comfort evaluation systems of shoe last are part of the process of personalized customization in footwear manufacturing. In addition to providing information on sports biology for footwear design, plantar pressure distribution can also predict diabetic foot changes and prevent plantar injuries. The last body's special structure and constraints must be considered when designing it with plantar pressure-sensing imaging data. Physiological and mechanical characteristics of the foot should be prioritized. Various fusion algorithms should be further enhanced in future research. To study healthy gaits, plantar pressure data from the research will be used to analyse human structure, function, and posture control. The use of dynamic image analysis of plantar pressure for rehabilitation can be further developed into clinical applications. A variety of diseases, such as obesity, diabetes, Parkinson's disease, and others, can affect plantar pressure. Research on functional shoe customization and precise last-making technology will provide new opportunities for research and impetus to the traditional shoemaking industry in the future.

While our research contributes to advancements in image segmentation and footwear design, we recognize that there are certain constraints and areas for improvement. Some limitations of our model include dataset bias and generalization to different sensor image datasets. Although we demonstrate the effectiveness of our model on plantar pressure images, its generalizability to other sensor image datasets should be further explored and validated. Different sensor types or data acquisition techniques may introduce variations that can impact the model's performance. The third is the computational complexity: The proposed ML-GNN model, combined with VGG16 and CNN layers, may have higher computational requirements compared to simpler segmentation models. This increased complexity could limit its applicability in resource-constrained environments or real-time applications. Finally, it is the subjectivity of image segmentation evaluation. Image segmentation evaluation can be subjective due to the absence of ground truth annotations that precisely capture all design elements. While we employ standard image segmentation indices to measure effectiveness, there may still be a subjective aspect to the evaluation process.

References

[1]. D. C. Deselnicu, A. M. Vasilescu, A. Mihai, A. A. Purcarea, and G. Militaru, "New products development through customized design based on customers' needs. Part 1: Footwear comfort parameters," *Procedia Technology*, vol. 22, pp. 1043–1050, 2016, doi: 10.1016/j.protcy.2016.01.148.

[2]. H.-C. Lu and W.-C. Chien, "Reshaping the wearing condition of high-heeled shoes by insole design to avoid hallux valgus," *Int. J. Industrial Ergonomics*, vol. 90, 103299, 2022, doi: 10.1016/j.ergon.2022.103299.

[3]. C. A. Abbott, K. E. Chatwin, P. Foden, A. N. Hasan, C. Sange, S. M. Rajbhandari, P. N. Reddy, L. Vileikyte, F. L. Bowling, A. J. M. Boulton, and N. D. Reeves, "Innovative intelligent insole system reduces diabetic foot ulcer recurrence at plantar sites: a prospective, randomized, proof-of-concept study," *The Lancet Digital Health*, vol. 1, no. 6, pp. e308–e318, Oct. 2019, doi: 10.1016/S2589-7500(19)30128-1.

[4]. D. Wang, Z. Li, N. Dey, A. S. Ashour, L. Moraru, R. S. Sherratt, and F. Shi, "Deep-segmentation of plantar pressure images incorporating fully convolutional neural networks," *Biocybernetics and Biomedical Engineering*, vol. 40, no. 1, pp. 546–558, Jan. 2020, doi: 10.1016/j.bbe.2020.01.004.

[5]. J. He, X. He, M. Zhang, S. Xiong, and H. Chen, "Deep dual-domain semiblind network for compressed image quality enhancement," *Knowledge-Based Systems*, vol. 238, 107870, Feb. 2022, doi: 10.1016/j.knosys.2021.107870.

[6]. A. Zotin, "Fast algorithm of image enhancement based on multiscale retinex," *Procedia Computer Science*, vol. 131, pp. 6–14, 2018, doi: 10.1016/j.procs.2018.04.179.

[7]. M. C. Hanumantharaju, M. Ravishankar, D. R. Rameshbabu, and S. Ramachandran, "Color image enhancement using multiscale retinex with modified color restoration technique," in *Proc EAIT*, Kolkata, India, 2011, pp. 93–97, doi: 10.1109/EAIT.2011.64.

[8]. S. Mittal, M. K. Dutta, and A. Issac, "Nondestructive image processing based system for assessment of rice quality and defects for classification according to inferred commercial value," *Measurement*, vol. 148, 106969, Dec. 2019, doi: 10.1016/j.measurement.2019.106969.

[9]. G. Tanisik, C. Zalluhoglu, and I. C. Nazili, "Multistream pose convolutional neural networks for human interaction recognition in images," *Signal Processing: Image Communication*, vol. 95, 116265, Jul. 2021, doi: 10.1016/j.image.2021.116265.

[10]. S. Baroud, S. Chokri, S. Belhaous, and M. Mestari, "A brief review of graph convolutional neural network based learning for classifying remote sensing images," *Procedia Computer Science*, vol. 191, pp. 349–354, 2021, doi: 10.1016/j.procs.2021.07.047.

[11]. M. Bongini, L. Rigutini, and E. Trentin, "Recursive neural networks for density estimation over generalized random graphs," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 11, pp. 5441–5458, Nov. 2018, doi: 10.1109/TNNLS.2018.2803523.

[12]. M. Gori, G. Monfardini, and F. Scarselli, "A new model for learning in graph domains," in *Proc. IJCNN*, Montreal, QC, Canada, 2005, pp. 729–734, doi: 10.1109/IJCNN.2005.1555942.

[13]. F. Scarselli, M. Gori, A. C. Tsoi, M. Hagenbuchner, and G. Monfardini, "The graph neural network model," *IEEE Trans. Neural Netw.*, vol. 20, no. 1, pp. 61–80, Jan. 2009, doi: 10.1109/TNN.2008.2005605.

[14]. M. M. Bronstein, J. Bruna, Y. LeCun, A. Szlam and P. Vandergheynst, "Geometric deep learning: Going beyond Euclidean data," *IEEE Signal Process. Mag.*, vol. 34, no. 4, pp. 18–42, Jul. 2017, doi: 10.1109/MSP.2017.2693418.

[15]. Y. Ding, Z. Zhang, X. Zhao, D. Hong, W.Cai, C. Yu, N. Yang, and W. Cai, Multifeature fusion: Graph neural network and CNN combining for hyperspectral image classification, Neurocomputing, vol. 501, pp. 246-257, 2022, doi.org/10.1016/j.neucom.2022.06.031.

[16]. A. Keramatfar, M. Rafiee, H. Amirkhani, Graph Neural Networks: A bibliometrics overview, Machine Learning with Applications, Vol. 10, 100401, 2022, doi.org/10.1016/j.mlwa.2022.100401.

[17]. S. Ding, C. Dong, T. Zhao, L. Koh, X. Bai, and J. Luo, "A meta-learning based multimodal neural network for multistep ahead battery thermal runaway forecasting," *IEEE Trans. Ind. Informat.*, vol. 17, no. 7, pp. 4503–4511, Jul. 2021, doi: 10.1109/TII.2020.3015555.

[18]. H. Li, Y. Cen, Y. Liu, X. Chen, and Z. Yu, "Different input resolutions and arbitrary output resolution: A meta learning-based deep framework for infrared and visible image fusion," *IEEE Trans. Image Process.*, vol. 30, pp. 4070–4083, Apr. 2021, doi: 10.1109/TIP.2021.3069339.

[19]. Q. Liu, X. Zhang, Y. Liu, K. Huo, W. Jiang, and X. Li, "Multipolarization fusion few-shot HRRP target recognition based on meta-learning framework," *IEEE Sensors J.*, vol. 21. no. 16, pp. 18085–18100, Aug. 2021, doi: 10.1109/JSEN.2021.3085671.

[20]. A. A. Salah and Y. Al-Salqan, "Meta-learning evolutionary artificial neural networks: by means of cellular automata," in *Proc CIMCA-IAWTIC*, Vienna, Austria, 2005, pp. 186–192, doi: 10.1109/CIMCA.2005.1631263.

[21]. B. Chen, J. Li, G. Lu, H. Yu, and D. Zhang, "Label co-occurrence learning with graph convolutional networks for multilabel chest X-ray image classification," *IEEE J. Biomed. Health Inform.*, vol. 24, no. 8, pp. 2292–2302, Aug. 2020, doi: 10.1109/JBHI.2020.2967084.

[22]. Y. Diao, J. Chen, and Y. Qian, "Multilabel remote sensing image classification with deformable convolutions and graph neural networks," in *Proc. IGARSS*, Waikoloa, HI, USA, 2020, pp. 521–524, doi: 10.1109/IGARSS39084.2020.9324530.

[23]. D. Lin, J. Lin, L. Zhao, Z. J. Wang, and Z. Chen, "Multilabel aerial image classification with a concept attention graph neural network," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, pp. 1–12, Mar. 2021, doi: 10.1109/TGRS.2020.3041461.

[24]. L. Studer, J. Wallau, R. Ingold, and A. Fischer, "Effects of graph pooling layers on classification with graph neural networks," in *Proc. SDS*, Luzern, Switzerland, 2020, pp. 57–58, doi: 10.1109/SDS49233.2020.00021.

[25]. D. Valsesia, G. Fracastoro, and E. Magli, "Deep graph-convolutional image denoising," *IEEE Trans. Image Process.*, vol. 29, pp. 8226–8237, Aug. 2020, doi: 10.1109/TIP.2020.3013166.

[26]. T. Hospedales, A. Antoniou, P. Micaelli and A. Storkey, "Meta-learning in neural networks: a survey" *IEEE Transactions on Pattern Analysis & Machine Intelligence*, vol. 44, no. 09, pp. 5149-5169, 2022. doi: 10.1109/tpami.2021.3079209

[27]. A. Hatamizadeh, B.Sedaee, Simulation of carbonate reservoirs acidizing using machine and meta-learning methods and its optimization by the genetic algorithm, Geoenergy Science and Engineering, vol. 223, 211509, 2023 doi.org/10.1016/j.geoen.2023.211509.
[28]. S. Huang, W. Gu, and S. Chen, “Optimization of classification rules and voting strategies for random forest,” in *Proc. CCIS*, Xi'an, China, 2021, pp. 381–387, doi: 10.1109/CCIS53392.2021.9754599.
[29]. M. M. Kalayeh and M. Shah, “On symbiosis of attribute prediction and semantic segmentation,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 5, pp. 1620–1635, May 2021, doi: 10.1109/TPAMI.2019.2956039.
[30]. Olaf Ronneberger, Philipp Fischer, Thomas Brox, U-Net: Convolutional Networks for Biomedical Image Segmentation, arXiv:1505.04597, 2015.
[31]. V. Badrinarayanan, A. Kendall, and R. Cipolla, “SegNet: A deep convolutional encoder-decoder architecture for image segmentation,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 12, pp. 2481–2495, Dec. 2017, doi: 10.1109/TPAMI.2016.2644615.
[32]. L. C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, “Semantic image segmentation with deep convolutional nets and fully connected CRFs,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 4, pp. 834–848, Apr. 2018, doi: 10.1109/TPAMI.2017.2699184.
[33]. Lafferty, J., McCallum, A., Pereira, F., Conditional random fields: Probabilistic models for segmenting and labelling sequence data". Proc. 18th International Conf. on Machine Learning. Morgan Kaufmann. 2011, pp. 282–289.
[34]. H. Zhao, J. Shi, X. Qi, X. Wang and J. Jia, "Pyramid Scene Parsing Network," *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Honolulu, HI, USA, 2017, pp. 6230-6239, doi: 10.1109/CVPR.2017.660
[35]. J. Zhou, M. Hao, D. Zhang, P. Zou, and W. Zhang, “Fusion PSPnet image segmentation based method for multifocus image fusion,” *IEEE Photon. J.*, vol. 11, no. 6, pp. 1–12, Dec. 2019, doi: 10.1109/JPHOT.2019.2950949.
[36]. E. Rivas-Posada and M. I. Chacon-Murguia, “General meta-learning paradigm based on prior-models, meta-model, meta-algorithm, and few-shot-base-model,” in *Proc. IJCNN*, Shenzhen, China, 2021, pp. 1–8, doi: 10.1109/IJCNN52387.2021.9533374.
[37]. Z. Li, D. Wang, N. Dey, A. S. Ashour, R. S. Sherratt, and F. Shi, “Plantar pressure image fusion for comfort fusion in diabetes mellitus using an improved fuzzy hidden Markov model,” *Biocybernetics and Biomedical Engineering*, vol. 39, no. 3, pp. 742–752, Jul. 2019, doi: 10.1016/j.bbe.2019.06.007.
[38]. Bishop, C. M., & Nasrabadi, N. M. Pattern recognition and machine learning, vol. 4, no. 4, pp. 738), 2016. doi.org/10.7551/mitpress/13811.003.0006
[39]. M. Zhou, W. Jiang, J. Wang, Design and manufacture of intelligent fabric-based insoles for disease prevention by monitoring plantar pressure, Materials Today Communications, vol. 37, 2023, pp.107646, https://doi.org/10.1016/j.mtcomm.2023.107646.
[40]. E. Pintelas, I. E. Livieris, S. Kotsiantis, P.s Pintelas, A multi-view-CNN framework for deep representation learning in image classification, Computer Vision and Image Understanding, Vol. 232, 2023, pp. 103687, https://doi.org/10.1016/j.cviu.2023.103687.

Acknowledgement:

This work was supported by Wenzhou Science and Technology Bureau Public Science and Technology Project: Personalized Shoe-Last Essence Design Technology for Women’s Fashion Shoes under grant no. G20160038.

Dan Wang received a PhD degree from Tianjin Key Laboratory of Process Measurement and Control, School of Electrical Engineering and Automation, Tianjin University. She is now an Assistant Professor at Wenzhou Polytechnic. Dr. Wang has published over 20 papers and conference proceedings; her research interests include optical pressure measurement, data analysis and image processing.

Zairan Li received his PhD degree from Tianjin Key Laboratory of Process Measurement and Control, School of Electrical Engineering and Automation, Tianjin University. Dr Li is now a Full Professor at Wenzhou Polytechnic. Dr. Li published over 40 papers and conference proceedings; his research interests include fuzzy logic, image mining, industrial design, intelligent design machine learning, biomedical imaging technologies and artificial intelligence in industrial applications.

Nilanjan Dey is an Associate Professor in the Department of Computer Science and Engineering, Techno International New Town, Kolkata, India. He is a visiting fellow of the University of Reading, UK. He also holds a position of Adjunct Professor at Ton Duc Thang University, Ho Chi Minh City, Vietnam. Previously, he held an honorary position of Visiting Scientist at Global Biomedical Technologies Inc., CA, USA (2012–2015). He was awarded his PhD from Jadavpur University in 2015. He is the Editor-in-Chief of the *International Journal of Ambient Computing and Intelligence*, IGI Global, USA. He is the Series Co-Editor of Springer *Tracts in Nature-Inspired Computing* (Springer Nature), Data-Intensive Research (Springer Nature), Advances *in Ubiquitous Sensing Applications for Healthcare* (Elsevier). He is an associate editor of IET Image Processing and editorial board member of Complex & Intelligent Systems, Springer Nature, Applied Soft Computing, Elsevier, etc. He is the Indian Ambassador of the International Federation for Information Processing-Young ICT Group, Fellow of IETE and Senior member of IEEE.

Rubén González Crespo has a PhD in Computer Science Engineering. Currently, he is Vice-Chancellor of Academic Affairs and Faculty from UNIR and Global Director of Engineering Schools from PROEDUCA Group. He is an advisory board member for the Ministry of Education at Colombia and an evaluator from the National Agency for Quality Evaluation and Accreditation of Spain (ANECA). He is a member of different committees at the ISO Organization. Finally, he has published more than 200 papers in indexed journals and congresses.

Fuqian Shi (Senior Member, IEEE) graduated from the College of Computer Science and Technology, Zhejiang University, received his PhD in Engineering and was a visiting Associate Professor at the Department of Industrial Engineering and Management System, University of Central Florida, USA from 2012 to 2014. He is a Senior Member of IEEE, Membership of ACM; and serves as over 50 committee board members of international conferences; Dr. Shi also serves as an Associate Editors of the International Journal of Ambient Computing and Intelligence (IJACI), International Journal of Rough Sets, and Data Analysis (IJRSDA). Dr. Fuqian published over 120 journal papers and conference proceedings; his research interests include fuzzy inference systems, artificial neural networks, biomechanical engineering, and bioinformatics.

R. Simon Sherratt (Fellow, IEEE) received a B.Eng. from Sheffield City Polytechnic in 1992, M.Sc. from The University of Salford in 1993, and PhD from The University of Salford in 1996. In 1996, he was appointed as a Lecturer in Electronic Engineering with the University of Reading, where he is currently a Professor of Biosensors. His research area is wearable devices, mainly for healthcare and emotion detection.

Eur Ing Professor Sherratt was awarded the 1st place IEEE Chester Sall Award in 2004, 2nd place in 2014, 3rd place in 2015 and 3rd place in 2016 for best papers in the IEEE TRANSACTIONS ON CONSUMER ELECTRONICS.