

Optimization of novel feature extraction for foot strike pattern recognition

Ponglert Sangkaphet¹, Wanida Kanarkard^{1*}, Wiroj Taweepworadej¹, Kitt Tientanopajai¹ and Kittipong Pengsri²

¹ Department of Computer Engineering, Faculty of Engineering, Khon Kaen University, Khon Kaen 40000, Thailand

² Department of Exercise and Sports Science, Faculty of Graduate School, Khon Kaen University, Khon Kaen 40000, Thailand

ABSTRACT

***Corresponding author:**
Wanida Kanarkard
wanida@kku.ac.th

Received: 26 May 2022
Revised: 10 November 2022
Accepted: 14 November 2022
Published: 19 December 2022

Citation:
Sangkaphet, P., Kanarkard, W., Taweepworadej, W., Tientanopajai, K., and Pengsri, K. (2022). Optimization of novel feature extraction for foot strike pattern recognition. *Science, Engineering and Health Studies*, 16, 22020007.

Foot strike pattern has a massive effect on the knee joint of the runner. An incorrect pattern while running can hurt the runner and significantly decrease running performance. The strike index is the most popular approach used to detect the strike pattern of a runner. However, this method requires expensive equipment in a laboratory environment, which creates difficulty for the experiment and significant costs. The purpose of this study paper was to develop a system to detect foot strike patterns during running using an inexpensive wireless wearable sensor system using hybrid center of pressure and principal component analysis for feature generation and machine learning for pattern classification. Furthermore, different classifiers were compared, to determine the optimal classifier. As a result, the proposed method improved performance in machine learning for foot strike pattern classification; the best classifier was support vector machine (radial basis function), which offered accuracy of 98.68%. This recognition system was thus established and able to successfully detect foot strike patterns. With this system, runners can adjust their foot strike pattern to achieve optimal results.

Keywords: foot strike pattern recognition; machine learning; smart wireless wearable sensor system

1. INTRODUCTION

Running is the most popular form of exercise, as it does not require equipment, and is a basic human movement. Normally, running techniques, known as foot strike patterns, can be classified into three categories: rearfoot strike (RFS), where the heel of the foot first strikes the ground; midfoot strike (MFS), where the ball and heel hit the ground at similar times; and forefoot strike (FFS), where the ball of the foot first strikes the ground. About 75% of runners use RFS, 24% use MFS, and the remaining 1% use FFS (Hasegawa et al., 2007; Kerr et al., 1983). RFS has a higher impact load than FFS, which leads to more knee injuries. However, FFS causes greater strain on the

calf and hamstring compared to RFS, which may cause Achilles tendinopathy. Therefore, one way to avoid injury is, for runners, to frequently change their foot strike patterns (Kulmala et al., 2013), which is not an easy task (Almeida et al., 2015).

Currently, foot strike patterns use the strike index (SI) for detection. The SI involves calculation of the center of pressure (CoP) on the plantar during an initial touch to the ground. The location of the CoP is then calculated along the longitudinal axis of the foot as a percentage of total foot length. The SI in the range of 0-33% is defined as RFS, 34%-66% is MFS, and 67%-100% is FFS (Cavanagh and Lafortune, 1980). The SI method requires a force plate or force treadmill in a laboratory setting. Therefore, foot

strike pattern detection outside of a laboratory is difficult and costly (Cheung et al., 2017). Furthermore, the occurrence of noise on a force plate can reduce the efficiency of CoP calculation accuracy (Altman and Davis, 2012).

Altman and Davis (2012) used an 8-camera Nexus system with a high framerate to achieve the kinematic measurement of a foot strike angle (FSA). The FSA was measured from the angle of the foot at foot strike, subtracting the angle of the foot while standing. In an experiment, SI and FSA measurements were concurrently collected. The relationship between SI and FSA was then established using linear regression. Experimental results showed that FSA can be represented by the SI. Giandolini et al. (2014) identified foot strike patterns during running by using the measurement of time between the heel and metatarsal acceleration peaks (THM) (Giandolini et al., 2014). THM was measured from two accelerometer sensors mounted on the shoes. The first sensor was affixed at the heel, and one more was affixed at the metatarsal. THM measurements were concurrently collected along with the FSA. Bravais-Pearson and Spearman correlation coefficients were used for the relationship analysis between THM and FSA in foot strike classification. Cheung et al. (2017) used low-cost force sensors for foot strike pattern detection to measure the onset time difference (OTD) (Cheung et al., 2017). OTD used two force sensing resistor (FSR) sensors mounted on the insole. The first sensor was affixed at the toe, and the other was affixed at the heel. The OTD measurement referred to the timing and location of FSR sensors at the point of contact. In a previous experiment, SI and OTD were measured and analyzed via linear regression. The experimental results indicated that OTD measurements can be provided along with a surrogate measure of SI. Eskofier et al. (2013) proposed a method to classify foot strike patterns by using the measurement of an accelerometer sensor installed in the laces of the right shoe (Eskofier et al., 2013). Acceleration in the three axes was continuously recorded along with a high-speed video, which created the labels of the foot strike pattern classes. A support vector machine (SVM) was then used to classify foot strike patterns. Experimental results showed that the SVM classifier could classify foot strike patterns with 95.3% accuracy. Hegde and Sazonov (2014, 2015) proposed the use of SmartStep, which is a wearable sensor device consisting of a 3D accelerometer, 3D gyroscope, and resistive pressure sensor. SmartStep can be wirelessly connected to an android smartphone application with data recording and visualization capabilities. The authors also presented SmartStep 2.0, a versatile insole monitor that can wirelessly charge, which allows for the insole system to be used in different application scenarios.

Several studies developed a wireless wearable sensor system that uses insole sensors and an inertial measurement unit (IMU) to measure foot pressure along with a three-axis accelerometer and gyroscope. Measured data were used for posture and activity analysis via machine learning (Ramirez-Bautista et al., 2017). Sazonov et al. (2011) proposed a shoe-based wearable sensor device and pattern recognition methodology for six different postures and activities (Sazonov et al., 2011). The authors designed a wearable sensor device that integrated a pressure-sensitive insole and accelerometers. Acquired data were used for recognition with SVMs, and the results provided accuracy of 95% in posture and activity recognition. Furthermore, Antwi-Afari et al. (2018) proposed a method applying a wearable

sensor system for the fall risk detection of construction workers. The data of 10 volunteers were collected from wearable insole pressure sensors (Antwi-Afari et al., 2018). Afterwards, machine-learning algorithms were used to learn the unique patterns of four loss-of-balance events (slip, trip, unexpected step-down, and twisted ankle). The study compared classification performance under various window sizes, feature groups, and types of classifiers. The experimental results showed that the best classification accuracy was achieved when using the random forest classifier. Ohnishi et al. (2019) proposed a method for recognizing 22 kinds of daily postures and gestures using foot pressure sensors. Furthermore, optimal positions for pressure sensors on the soles were investigated from the perspective of motion recognition accuracy, and the optimal measurement points for high recognition accuracy were examined by evaluating combinations of two and three foot pressure measurement areas on a round-robin basis. The experimental results showed that, in all areas, the best classifier was random forest with a classification accuracy of 99.7%. Random forest was used in cases of two and three optimal areas. When selecting the two optimal points for all subjects, recognition accuracy was about 91.9% on average, and average recognition accuracy in a three-point evaluation was 98.4%. Zhao et al. (2019) proposed gait detection based on foot-mounted inertial sensors, which detected the following gait patterns: heel strike, foot flat, midstance, heel off, toe off, and midswing (Zhao et al., 2019). The hidden Markov model and a neural network were used in solving the problem of gait detection. Mei et al. (2020) classified foot types into normal feet, cavus feet, and planus feet using sensor insoles combined with 1D-convolutional neural networks (1D-CNNs). Experimental results showed that the highest classification accuracy was 99.26%.

This paper aimed to optimize foot strike pattern recognition via novel feature extraction. A novel and inexpensive smart wireless wearable sensor system was designed and installed inside running shoes for the real-time collection of foot strike patterns. The foot strike pattern data were pre-processed with a hybrid CoP and principal component analysis (PCA) for feature extractions before classifying the running pattern via machine learning. The different classifiers were compared and consisted of k-nearest neighbor (KNN), SVM (linear kernel function), SVM (polynomial kernel function), SVM (radial basis function (RBF) kernel function), artificial neural network (ANN), and random forest (RF) to find the best classifier.

2. MATERIALS AND METHODS

2.1 Hardware description

Figure 1 shows the hardware components. Twelve FSR sensors are illustrated in Figure 2a. Each FSR sensor consisted of a conductive polymer that predictably changed its resistance after applying a force to its surface. The greater the force, the lower the resistance. When no force was applied to the FSR, its resistance was larger than 1 M Ω . The FSRs used here had a round 20 mm diameter sensing area (Interlink Electronics, 2019). Twelve FSR sensors were placed onto each insole: one sensor was located at the hallux (toe); three sensors were located at the forefoot (first, third, and fifth metatarsals); five sensors were located at the paw; and three sensors were located at the heel. The 12 FSR sensors were connected with a CD74HC4067 multiplexer module.

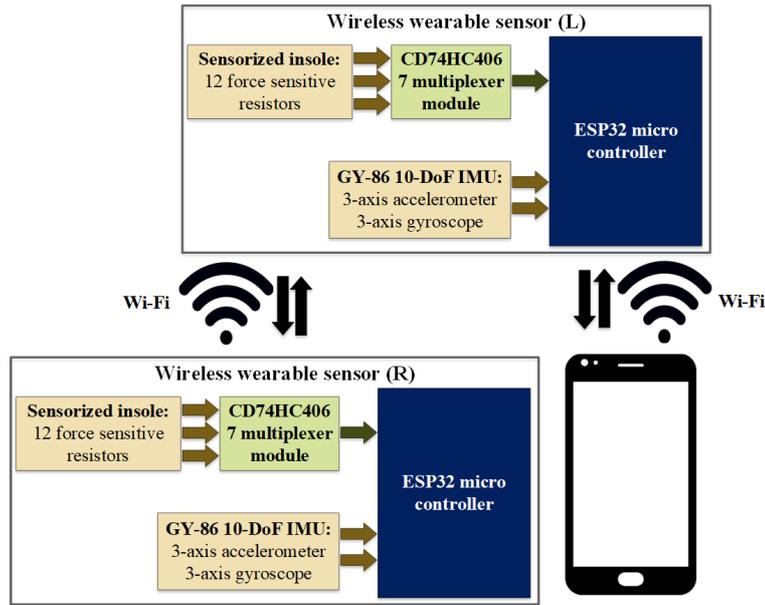


Figure 1. Hardware components

Figure 2c illustrates the prototype. The ESP32 microcontroller and CD74HC4067 multiplexer module were encapsulated inside a plastic box, which was attached to the lateral side of the tibia. A 3.7 V lithium-ion rechargeable battery was used to power the circuit board (Figure 2b), which was firmly affixed to the lateral side of the plastic box. The 10 DOF IMU, GY86, is a small board and was attached to the laces of the running shoes (Figure 2c). GY86 was used in Euler angle calculations and could connect to the ESP32 module through the I2C protocol. There were four sensors on the GY86: an accelerometer (MPU6050), a gyroscope (MPU6050), a magnetometer

(HMC5883L), and a barometer and temperature sensor (MS5611). The ESP32 Wi-Fi of the left shoe acted as an access point for the system to create its own Wi-Fi network, to which the nearby ESP32 Wi-Fi of the right shoe and the Wi-Fi of the smartphone could be connected.

The signals from the FSR and IMU of both shoes were processed by the ESP32 microcontroller and packaged to be sent to the smartphone via the ESP32 Wi-Fi. The sensor data from both shoes were sent to the application on the smartphone with a sampling rate of 25 Hz for running monitoring, which was achieved on the Android platform.

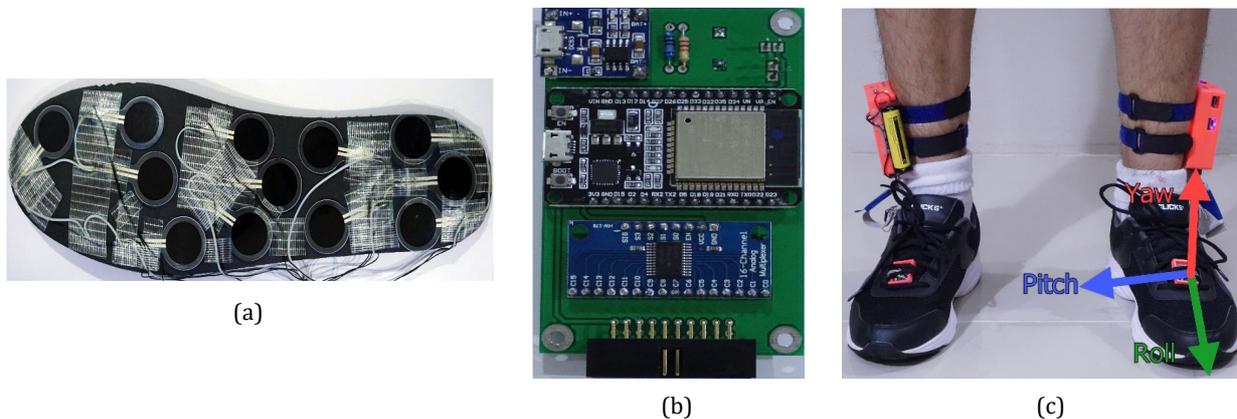


Figure 2. (a) Placement of twelve FSR sensors underneath the insole, (b) circuit board, and (c) prototype and Euler angle in the left foot

2.2 Data collection

A total of 16 participants (14 males and 2 females; age range = 19–33 years; mean age = 20.9 ± 2.6 years; height = 172.5 ± 5.6 cm; weight = 64.4 ± 8.0 kg; exercise 2–5 times/week) were recruited for data collection. The shoe size (US) of adult males ranged from 8 to 10, and that of adult females ranged from 5 to 7. All participants signed an informed consent form, and the experiment was

approved by the Center for Ethics in Human Research, Khon Kaen University (HE632050). All participants were asked to run on an instrumented treadmill with a running speed of 6.5–7 km/h. Each participant ran under three conditions, RFS, MFS, and FFS, with instruction from the expert to assist MFS and FFS running by correcting the participant's body posture. Each running condition lasted for 2 min 30 s with a 2-min break between conditions.

2.3 Foot strike pattern recognition using machine learning

In this article, foot strike patterns during running were grouped into one of three classes: RFS, MFS, and FFS. The proposed foot strike pattern recognition shown in

Figure 3 was based on three steps: 1) CoP feature extraction; 2) feature selection (PCA algorithm); and 3) classification of foot strike patterns (where the multiclassifier algorithms of KNN, SVM, ANN, and RF were compared.

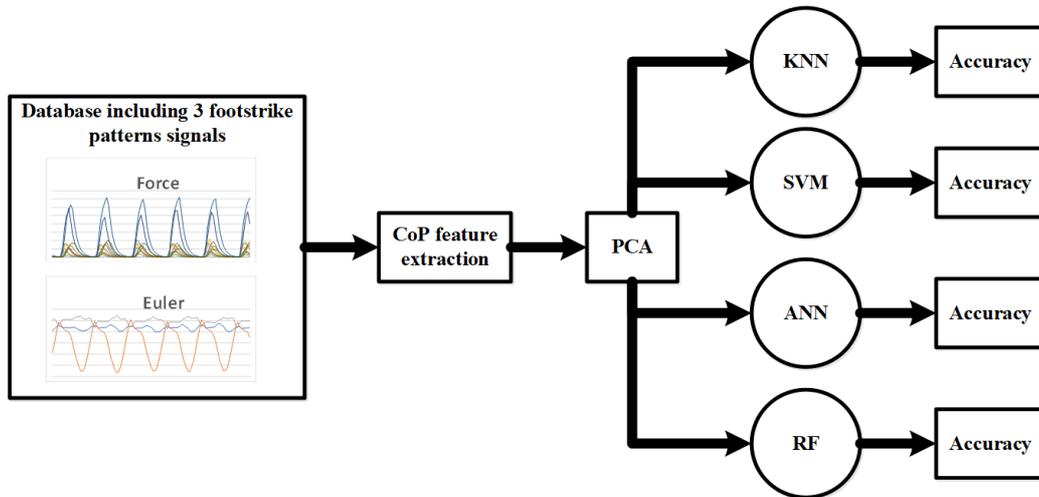


Figure 3. Proposed foot strike pattern recognition algorithm

2.3.1 CoP feature extraction

Raw data from the wireless wearable sensor system included FSR values and Euler angles and were transformed into CoP feature vectors. FSR values were calculated as the CoP (Cavanagh and Lafortune, 1980), and Euler angles were normalized. CoP feature vectors represented the CoP

trajectories of the foot strike pattern. CoP feature extraction consisted of the following three steps.

First, sensor data were normalized to the scale of [0,1]. The placement coordinates of 12 FSR sensors were then configured by a physician (Figure 4), and Euler angles in the dataset were normalized.

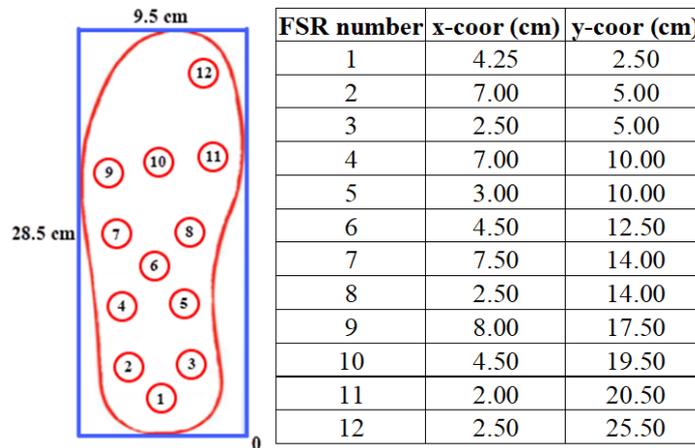


Figure 4. Placement of 12 FSR sensors and corresponding coordinates

Second, FSR values in the dataset were calculated as the CoP with the previously normalized placement coordinates of 12 FSRs. Equations (1) and (2) show the CoP calculations:

$$CoP_x = \frac{\sum_{i=1}^M F_i x_i}{\sum_{i=1}^M F_i} \quad (1)$$

$$CoP_y = \frac{\sum_{i=1}^M F_i y_i}{\sum_{i=1}^M F_i} \quad (2)$$

where CoP_x represents the CoP location along the transverse axis of the foot, CoP_y represents the CoP location along the longitudinal axis of the foot, $M = 12$ is the number of FSR sensors, F_i is the i output of FSR sensors, x_i is the normalized i transverse axis coordinates of the FSR sensors, and y_i is the normalized i longitudinal axis coordinates of the FSR sensors. Lastly, the CoP values and Euler angle time series were transformed into CoP feature vectors with a time interval of 1 s. The data sample of a single shoe was represented by vector

$S = \{CoP_x, CoP_y, \emptyset, \theta, \varphi\}$ where CoP_x, CoP_y represent the CoP location along the transverse and longitudinal axis of the foot, respectively; \emptyset is roll; θ is pitch; and φ is yaw. The time series of the data from both shoes were then integrated as $f_j = \{S_L, S_R\}_j, j = \{1, 2, 3, \dots, K\}$, where S_L, S_R are the data samples of the left and right shoe, respectively, and K is the number of samples in 1 s at a 25 Hz sampling frequency. Here, the CoP feature vector of the data samples in 1 s at the 25 Hz sampling frequency was $V = \{f_1, f_2, f_3, \dots, f_K\}$. Due to transformation, the size of the CoP feature vectors was 250 elements (2 shoes \times (2 coordinates of CoP + 3 axes of Euler angles) \times 25 samples/s \times 1 s = 250 elements). A 1 s duration was chosen for the samples, as this time interval was sufficient to characterize the foot strike patterns.

2.3.2 Feature selection: PCA for dimension reduction

CoP feature extraction generated high-dimensional features (size of CoP feature vectors = 250 dimensions), which significantly reduced system performance. Feature selection was used to filter irrelevant or redundant features from the dataset to improve model accuracy and achieve faster training of algorithms. PCA was used in this work for feature selection because of its low noise sensitivity and decreased requirements for capacity and memory. PCA projects the entire dataset onto a different feature subspace through the following steps (Smith, 2002). The first step involved computing the mean values in each dimension with Equation (3):

$$\bar{A} = \frac{1}{N} \sum_{i=1}^N a_i \quad (3)$$

where \bar{A} is the mean of the data, a_i is the i sample of the dataset, and N is the total number of samples of the dataset.

The second step was to compute the covariance matrix with Equation (4):

$$\text{cov} = \frac{1}{N-1} \sum_{i=1}^N (a_i - \bar{A})(a_i - \bar{A})^T \quad (4)$$

The third step was to compute the eigenvectors and eigenvalues from the covariance matrix with Equation (5):

$$(\text{Cov} - \lambda_j I)z_j = 0 \quad (5)$$

where $j = \{1, 2, 3, \dots, D\}$, $D = 250$ is the size of the CoP feature vectors, I is the identity matrix, λ_j is the eigenvalues, and z_j is the eigenvectors. Eigenvectors represent the variance directions, and eigenvalues represent the magnitude of variance.

The next step was to select components form a feature vector. Eigenvalues were sorted in descending order, and d eigenvectors that corresponded to the d largest eigenvalues were selected, where d is the number of components of the new feature subspace ($d < 250$). The approach was to select the number of components by using the explained variance calculated from the eigenvalues (Smith, 2002). The explained variance was calculated for the components to generate a cumulative summation of the explained variance plot in which the number of components selected from a cumulative summation of the explained variance at d components was slightly different. Furthermore, the projection matrix (W) was constructed from the d selected eigenvectors.

Lastly, for projection onto the new feature space, the CoP feature dataset was transformed into the new d -

dimensional subspace by multiplying the transpose of W as shown in Equation (6):

$$P = W^T a \quad (6)$$

where P is the d -dimensional matrix of the transformed CoP feature dataset.

2.3.3 Pattern recognition models of foot strike patterns

To classify different types of foot strike patterns, machine-learning classifiers were used to learn unique sensor data patterns from foot plantar pressure data and Euler angles based on CoP feature extraction. We investigated the best classification methods, including KNN, SVM, RF, and ANN. Furthermore, for SVM, the kernel function was modified to include three kernels: linear kernel (SVM linear function), polynomial kernel (SVM poly function), and radial basis function kernel (SVM RBF function).

KNN is a lazy learning algorithm. When an unknown data class is received, the algorithm compares similarities with the closest k number of instances, and the unknown data class is assigned a class label based on the most classes of k instances. KNN is performed selecting parameter k , calculating the distance metric between a new data point and the entire sample dataset, sorting the sample dataset in ascending order by distance and choosing the lowest-distance k data samples. In this work, Euclidean distance was used as the distance metric (Kataria and Singh, 2013).

RF is an ensemble classifier that generates a model from several decision trees, each of which is trained using a different subset of the entire dataset. Afterwards, each decision tree model calculates the prediction results to choose the decision tree with the best prediction. RF is performed by feeding the input dataset into each tree in the RF, to training each tree, and aggregating the prediction results of each tree with voting and selecting the tree with the highest number of votes (Mushtaq and Mellouk, 2017).

SVM is a classification technique used in pattern recognition. The principle of SVM is to separate the best hyperplanes between the data points of different classes. The training principle of SVM is to map a set of training data $D = \{x_i, y_i\}, i = 1, 2, 3, \dots, N$ where $x_i = (x_{i1}, x_{i2}, \dots, x_{im}) \in \mathbb{R}^m$, $y_i \in \{1, -1\}$, and N is the total number of samples in the training set. The optimization problem for SVM is outlined in Equations (7)-(9) (Hu et al., 2004):

$$\min_{w, \xi, b} J(w, \xi) = \frac{1}{2} \|w\|^2 + C \sum_{i=1}^N \xi_i \quad (7)$$

Thus,

$$y_i (w^T \varphi(x_i) + b) + \xi_i - 1 \geq 0, i = 1, \dots, N \quad (8)$$

$$\xi_i \geq 0 \quad i = 1, \dots, N \quad (9)$$

where C is a parameter selected by users that is a positive constant, w is the weight vector, ξ_i is a positive slack variable that represents the distance between x_i and the decision boundary, φ is a nonlinear function used to map input data point x_i into a higher dimensional space, and b is a bias value. However, Lagrange multipliers (β) are used in Equation (8). This model can be rewritten as Equation (10):

$$f(x) = \sum_{x_i \in SV} \beta_i y_i K(x, x_i) + b \quad (10)$$

where $K(x, x_i)$ is the kernel function. In this work, we used the polynomial and RBF kernels, defined as $K(x, x_i) = (\gamma x^T x_i + r)^d$

and $K(x, x_i) = \exp(-\gamma \|x - x_i\|^2)$, respectively, where γ , r , and d are the kernel parameters. The SVM classifier implemented for the multiclassification problem was one-against-one.

ANN is a popular machine-learning technique that is widely used for pattern recognition. The most popular neural network algorithm is the multilayer backpropagation neural network, which consists of three layers: an input layer, a hidden layer, and an output layer. In this work, a multilayer backpropagation neural network was used, with an input layer containing the number of input nodes according to the size of the CoP feature vectors, an output layer with three output nodes, and a hidden layer to control parameters ranging from 3 to 250 nodes. The activation functions of the hidden and an output layer were ReLU and softmax, respectively.

For performance assessment, sensor data were used to divide the sample data into training and testing sets. In total, 80% of the sample sensor data from 2-min trials of all subjects and running conditions were used for tenfold cross-validation to find the optimal parameters and build the classifier. Twenty percent of the sample sensor data from 30-s trials of all subjects and running conditions were reserved for validation testing. Tenfold cross-validation was applied to divide the data into ten sections. Nine sections were used as the training set, and one section was used as the testing set. This process was repeated until each of the ten sections was used as the testing set.

Accuracy was calculated from the average accuracy across tenfold cross-validation.

A grid search approach was used to find the optimal parameters of classifiers. The k parameter of KNN was tuned for $k \in \{3, 4, 5, \dots, 15\}$. The number of trees in the RF was tuned for $N_{\text{estimators}} \in \{100, 300, 500, 800, 1200, 1500, 2000\}$. The maximum depth of the RF tree was tuned for $\text{MaxDepth} \in \{5, 8, 15, 25, 30, 40\}$. The C parameter of the SVM was tuned for $C \in \{2^{-5}, 2^{-4}, 2^{-3}, \dots, 2^{15}\}$. The degree parameter of the polynomial kernel was tuned for $d \in \{2, 3, 4, 5\}$. The gamma parameter of the RBF kernel was tuned for $\gamma \in \{2^{-15}, 2^{-14}, 2^{-13}, \dots, 2^3\}$. The hidden size parameter of ANN was tuned for hidden sizes $\in \{3, 4, 5, \dots, 250\}$.

3. RESULTS AND DISCUSSION

In total, 180,000 samples were collected. Due to the decimation of CoP feature extraction, the number of the feature vectors totaled 7,200 samples that were used in the comparison experiments between CoP and the hybrid of CoP and PCA. Furthermore, all CoP feature vectors were calculated to generate a cumulative summation of the explained variance plot, as shown in Figure 5.

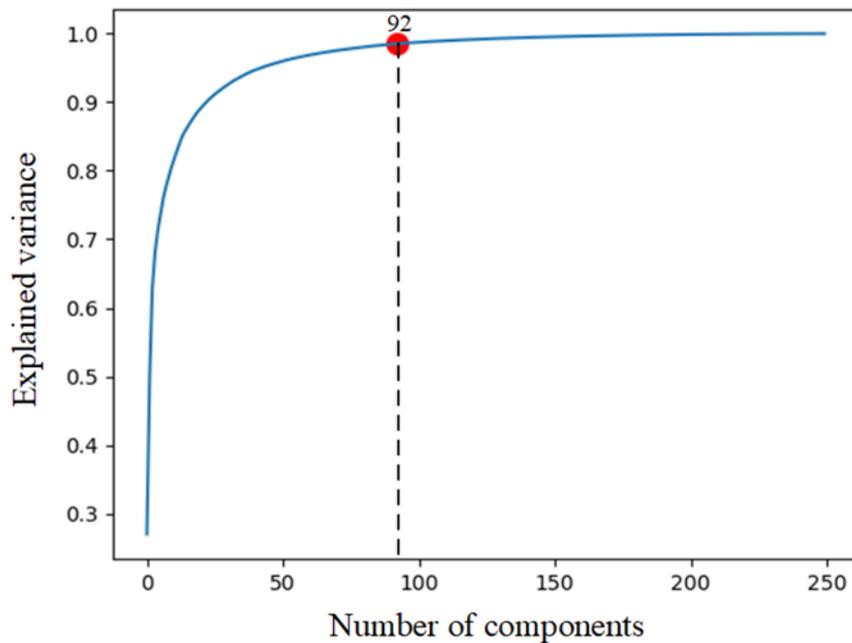


Figure 5. Relationship plot between number of components and explained variance of the CoP feature dataset

Figure 5 shows that 92 components were selected, since the explained variance of the 92 components was slightly different.

In this paper, the experiment was divided into two phases. First, the training set experiment with tenfold cross-validation was used to find the parameter that gave the best average accuracy for each classifier (KNN, SVM

(linear function), SVM (poly function), and SVM (RBF function), ANN, and RF), as shown in Table 1. Second, for the testing set for validation testing with the five different classifiers, a model was built from the training set using the optimal parameters. A confusion matrix of the six different classifier models from validation testing is presented in Table 2.

Table 1. Comparison performance of the six classification models

Classifier	Optimal parameter	Feature	Number of features	Accuracy (%)	Time (s)
KNN	k : 3	COP	250	72.11±10.0	1.4199
		COP + PCA	92	72.15±10.0	0.3276
SVM linear function	C : 0.0625	COP	250	66.78±13.3	7.6297
		COP + PCA	92	66.90±13.1	2.9655
SVM poly function	{C, degree}: {32768, 5}	COP	250	82.23±11.1	3.1608
		COP + PCA	92	78.54±9.9	1.5329
SVM RBF function	{C,γ}:{8, 0.125}	COP	250	85.03±8.9	3.2255
		COP + PCA	92	85.89±9.1	1.1884
ANN	hidden sizes: 236	COP	250	84.27±10.3	14.9785
		COP + PCA	92	85.03±8.5	5.6793
RF	{N_estimators}: {300} { MaxDepth } : { 30 }	COP	250	83.76±9.0	12.2358
		COP + PCA	92	75.78±10.3	10.5513

Table 1 presents the accuracy and time of the training set with tenfold cross-validation to find the optimal parameters. The best average accuracy for the parameter of classifiers was obtained with the following parameters: k = 3 in KNN, C = 0.0625 in SVM (linear function), C = 32,768 and d = 5 in

SVM (poly function), C = 8 and $\gamma = 0.125$ in SVM (RBF kernel function), hidden sizes = 236 in ANN, and N_estimators = 300 and MaxDepth = 30 in RF. The best method was found to be the SVM (RBF function) and CoP + PCA feature extraction, with an average accuracy of 85.89±9.1%.

Table 2. Confusion matrix of six different classifiers with CoP + PCA feature extraction

Classifier	Actual	Predicted				Recall
		FFS	RFS	MFS	Total of predicted	
KNN	FFS	467	6	7	480	0.9729
	RFS	5	445	30	480	0.9271
	MFS	9	32	439	480	0.9146
	Total of actual	481	483	476		
	Precision	0.9709	0.9213	0.9223		0.9382
SVM linear function	FFS	451	7	22	480	0.9396
	RFS	21	351	108	480	0.7313
	MFS	89	65	326	480	0.6792
	Total of actual	561	423	456		
	Precision	0.839	0.8298	0.7149		0.7833
SVM poly function	FFS	471	2	7	480	0.9813
	RFS	15	448	17	480	0.9333
	MFS	17	8	455	480	0.9479
	Total of actual	503	458	479		
	Precision	0.9364	0.9782	0.9499		0.9542
SVM RBF function	FFS	475	2	3	480	0.9896
	RFS	1	474	5	480	0.9875
	MFS	5	3	472	480	0.9833
	Total of actual	481	479	480		
	Precision	0.9875	0.9896	0.9833		0.9868
ANN	FFS	472	2	6	480	0.9833
	RFS	0	472	8	480	0.9833
	MFS	7	15	458	480	0.542
	Total of actual	479	489	472		
	Precision	0.9854	0.9652	0.9703		0.9736
RF	FFS	464	6	10	480	0.9667
	RFS	3	454	23	480	0.9458
	MFS	17	25	438	480	0.9125
	Total of actual	484	485	471		
	Precision	0.9587	0.9361	0.9299		0.9417

3.1 Convergence test

Table 2 presents the confusion matrix, showing the classification accuracy of KNN, SVM (linear function), SVM

(poly function), SVM (RBF function), ANN, and RF with hybrid CoP + PCA feature extraction based on validation testing. The numbers in italics show how much of the

predicted class was correct. Recall is the ratio of the correctly predicted classes, compared to the actual classes, and the sum of all predicted classes in a row. Precision is the ratio between the correctly predicted classes, compared to the

actual classes, and the sum of all predicted classes in a column. The confusion matrix in Table 2 showed that the lowest overall misclassification was achieved through the SVM (RBF function).

Table 3. Accuracy and time of validation testing

Classifier	Feature	Accuracy (%)	Time (s)
KNN	COP	93.75	1.3962
	COP + PCA	93.82	0.3051
SVM linear function	COP	77.77	1.1319
	COP + PCA	78.33	0.3411
SVM poly function	COP	97.77	0.4737
	COP + PCA	95.42	0.2303
SVM RBF function	COP	97.77	0.7619
	COP + PCA	98.68	0.2553
ANN	COP	97.15	0.0179
	COP + PCA	97.36	0.0069
RF	COP	96.66	0.0867
	COP + PCA	94.17	0.0787

Table 3 presents the accuracy and time of the testing set with six different classifier models built from the training set using optimal parameters. The best method was the SVM (RBF function) and CoP + PCA feature extraction with an accuracy rate of 98.68%.

The acquisition system in this paper consisted of 12 FSR sensors underneath the insole and IMU mounted onto the instep. The classification system distinguished between different foot strike patterns with a classification rate of 98.68%. The results showed that applying CoP feature extraction and PCA for new feature generation helped improve the efficiency of classifiers in solving the problem of foot strike pattern recognition. CoP feature extraction was based on CoP trajectory measurements under the plantar with Euler angle measurements on the instep over a given time period. These data were used to distinguish between the characteristics of various foot strike patterns. Furthermore, the dataset was decimated via CoP feature extraction, which generated high-dimensional features. Thereafter, PCA was applied for dimensional reduction, which reduced the number of dimensional features from 250 to 92. PCA helped filter irrelevant and redundant features from the dataset, resulting in classifier accuracy improvements, a smaller size, and faster training.

In addition, we compared the results in Table 1 for six different classifiers with hybrid CoP-PCA feature extraction. Tenfold cross-validation was used to find the optimal parameters and showed that the SVM (RBF function) provided the best average accuracy of $85.89 \pm 9.1\%$, followed by ANN, the SVM (poly function), RF, KNN, and the SVM (linear function), which provided average accuracy values of $85.03 \pm 8.5\%$, $78.54 \pm 9.9\%$, $75.78 \pm 10.3\%$, $72.15 \pm 10.0\%$, and $66.90 \pm 13.1\%$, respectively. Table 3 shows the case of validation testing, indicating that SVM (RBF function) provided the best accuracy with 98.68%, followed by ANN, SVM (poly function), RF, KNN, and SVM (linear function), with accuracy of 97.36%, 95.42%, 94.17%, 93.82%, and 78.33%, respectively. In both cases, the results showed that SVM (RBF function) was the best classifier. Furthermore, the results showed that hybrid CoP-PCA feature extraction was more accurate than CoP feature extraction for the SVM (RBF function), ANN, SVM (linear

function), and KNN. The accuracy of the RF and SVM (poly function) classifiers was lower.

The results in Table 2 showed that the proposed method achieved good precision and recall in the classification of foot strike patterns, while the SVM (RBF function) provided the lowest overall misclassification.

Overall, the proposed approach was able to classify the 3 types of foot strike patterns with good accuracy. In the comparison of the six different classifiers, SVM (RBF function) was the optimal classifier for foot strike pattern classification. This system is compact, wireless, and inexpensive and can be applied to several tasks, such as the real-time detection of foot strike patterns during running to adjust the foot strike pattern of the runner.

4. CONCLUSION

In this study, we developed and evaluated a novel optimization method using a hybrid of CoP and PCA in the feature extraction process for foot strike pattern recognition. A smart wireless wearable sensor system was designed using FSRs and IMU to collect the real time foot strike patterns, and these data were then sent to be classified using a machine learning model. The six machine learning models of KNN, SVM (linear function), SVM (poly function), SVM (RBF function), ANN, and RF were also examined to find the best classifier. The proposed model outperformed the unoptimized model, resulting in better accuracy and robustness. These promising results could help runners reduce injuries caused by using repetitive foot strike patterns over a long period of time while running. However, the measurement of foot strike patterns across various sizes, numbers, allocations of FSR sensors, and surface inclinations may afford different results, thus presenting a challenge for accurate recognition. This limitation could be studied in the future.

REFERENCES

Almeida, M. O., Davis, I. S., and Lopes, A. D. (2015). Biomechanical differences of foot-strike patterns during

- running: A systematic review with meta-analysis. *Journal of Orthopaedic & Sports Physical Therapy*, 45(10), 738-755.
- Altman, A. R., and Davis, I. S. (2012). A kinematic method for footstrike pattern detection in barefoot and shod runners. *Gait Posture*, 35(2), 298-300.
- Antwi-Afari, M. F., Li, H., Seo, J., and Wong, A. Y. L. (2018). Automated detection and classification of construction workers' loss of balance events using wearable insole pressure sensors. *Automation in Construction*, 96, 189-199.
- Cavanagh, P. R., and Lafortune, M. A. (1980). Ground reaction forces in distance running. *Journal of Biomechanics*, 13(5), 397-406.
- Cheung, R. T. H., An, W. W., Au, I. P. H., Zhang, J. H., Chan, Z. Y. S., Man, A., Lau, F. O. Y., Lam, M. K. Y., Lau, K. K., Leung, C. Y., Tsang, N. W., Sze, L. K. Y., and Lam, G. W. K. (2017). Measurement agreement between a newly developed sensing insole and traditional laboratory-based method for footstrike pattern detection in runners. *PLOS One*, 12(6), e0175724.
- Eskofier, B. M., Musho, E., and Schlarb, H. (2013). Pattern classification of foot strike type using body worn accelerometers. In *Proceedings of IEEE International Conference on Body Sensor Networks*, pp. 1-4. Cambridge, USA.
- Giandolini, M., Poupard, T., Gimenez, P., Horvais, N., Millet, G. Y., Morin, J. B., and Samozino, P. (2014). A simple field method to identify foot strike pattern during running. *Journal of Biomechanics*, 47(7), 1588-1593.
- Hasegawa, H., Yamauchi, T., and Kraemer, W. J. (2007). Foot strike patterns of runners at the 15-km point during an elite-level half marathon. *The Journal of Strength and Conditioning Research*, 21(3), 888-893.
- Hegde, N., and Sazonov, E. S. (2014). SmartStep: A fully integrated, low-power insole monitor. *Electronics*, 3(2), 381-397.
- Hegde, N., and Sazonov, E. S. (2015). SmartStep 2.0 - A completely wireless, versatile insole monitoring system. In *Proceedings of IEEE International Conference on Bioinformatics and Biomedicine*, pp. 746-749. Washington D.C., USA.
- Hu, Z. H., Li, Y. G., Cai, Y. Z., and Xu, X. M. (2004). An empirical comparison of ensemble classification algorithms with support vector machines. In *Proceedings of 2004 International Conference on Machine Learning and Cybernetics*, pp. 3520-3523. Shanghai, China.
- Interlink Electronics. (2019). *FSR force sensing resistor integration guide and evaluation part catalog*. [Online URL: <https://manuals.plus/fsr/fsr-force-sensing-resistor-integration-guide-and-evaluation-parts-catalog#axzz7nsi1Xxp2>] accessed on October 10, 2021.
- Kataria, A., and Singh, M. (2013). A review of data classification using k-nearest neighbour algorithm. *International Journal of Emerging Technology and Advanced Engineering*, 3(6), 354-360.
- Kerr, B. A., Beauchamp, L., Fisher, V., and Neil, R. (1983). Footstrike patterns in distance running. In *Proceedings of the International Symposium on Biomechanical Aspects of Sport Shoes and Playing Surfaces*, pp. 135-141. Alberta, Canada.
- Kulmala, J. P., Avela, J., Pasanen, K., and Parkkari, J. (2013). Forefoot strikers exhibit lower running-induced knee loading than rearfoot strikers. *Medicine and Science in Sports and Exercise*, 45(12), 2306-2313.
- Mei, Z., Ivanov, K., Zhao, G., Wu, Y., Liu, M., and Wang, L. (2020). Foot type classification using sensor-enabled footwear and 1D-CNN. *Measurement*, 165, 108184.
- Mushtaq, M. S., and Mellouk, A. (2017). Methodologies for subjective video streaming QoE assessment. In *Quality of Experience Paradigm in Multimedia Services* (Mellouk, S. ed.), pp. 27-57. London: ISTE Press.
- Ohnishi, A., Terada, T., and Tsukamoto, M. (2019). A method for recognizing postures and gestures using foot pressure sensors. *Journal of Information Processing*, 27, 348-358.
- Ramirez-Bautista, J. A., Huerta-Ruelas, J. A., Chaparro-Cardenas, S. L., and Hernandez-Zavala, A. (2017). A review in detection and monitoring gait disorders using in-shoe plantar measurement systems. *IEEE Reviews in Biomedical Engineering*, 10, 299-309.
- Sazonov, E. S., Fulk, G., Hill, J., Schutz, Y., and Browning, R. (2011). Monitoring of posture allocations and activities by a shoe-based wearable sensor. *IEEE Transactions on Biomedical Engineering*, 58(4), 983-990.
- Smith, L. I. (2002). A tutorial on principal components analysis. [Online URL: <https://ourarchive.otago.ac.nz/bitstream/handle/10523/7534/OUCS-2002-12.pdf>] accessed on October 13, 2021.
- Zhao, H., Wang, Z., Qiu, S., Wang, J., Xu, F., Wang, Z., and Shen, Y. (2019). Adaptive gait detection based on foot-mounted inertial sensors and multi-sensor fusion. *Information Fusion*, 52, 157-166.