Sensor-based Smart Home Technologies and Applications - Research Article

International Journal of
Distributed
Sensor Networks

International Journal of Distributed Sensor Networks
2018, Vol. 14(10)
© The Author(s) 2018
DOI: 10.1177/1550147718802186
journals.sagepub.com/home/dsn
SAGE

Golf swing classification with multiple deep convolutional neural networks

Libin Jiao[1,2], Rongfang Bie[1], Hao Wu[1], Yu Wei[3], Jixin Ma[4], Anton Umek[2] and Anton Kos[2]

Abstract

The use of smart sports equipment and body sensory systems supervising daily sports training is gradually emerging in professional and amateur sports; however, the problem of processing large amounts of data from sensors used in sport and discovering constructive knowledge is a novel topic and the focus of our research. In this article, we investigate golf swing data classification methods based on varieties of representative convolutional neural networks (deep convolutional neural networks) which are fed with swing data from embedded multi-sensors, to group the multi-channel golf swing data labeled by hybrid categories from different golf players and swing shapes. In particular, four convolutional neural classifiers are customized: "GolfVanillaCNN" with the convolutional layers, "GolfVGG" with the stacked convolutional layers, "GolfInception" with the multi-scale convolutional layers, and "GolfResNet" with the residual learning. Testing on the real-world swing dataset sampled from the system integrating two strain gage sensors, three-axis accelerometer, and three-axis gyroscope, we explore the accuracy and performance of our convolutional neural network–based classifiers from two perspectives: classification implementations and sensor combinations. Besides, we further evaluate the performance of these four classifiers in terms of classification accuracy, precision–recall curves, and F1 scores. These common classification indicators illustrate that our convolutional neural network–based classifiers can basically group the golf swing predefined by the combination of shapes and golf players correctly and outperform support vector machine method representing traditional classification methods.

Keywords

Smart device, golf data analysis, classification, convolutional neural network, deep learning

Date received: 11 December 2017; accepted: 23 August 2018

Handling Editor: Jui-Pin Yang

Introduction

Advances in technology and data science are changing the way of practicing and training in recreational, amateur, and professional sports. The collection of sports performance data has become easier and more reliable with the development of miniature, lightweight sensors, sensor networks, and communication technologies. The key issue now is how to analyze the large amounts of (streaming) data from the above-mentioned wearable devices. The processing requirements for sensor signals and data have become more demanding, both in volume and time constraints.

[1] College of Information Science and Technology, Beijing Normal University, Beijing, China
[2] Faculty of Electrical Engineering, University of Ljubljana, Ljubljana, Slovenia
[3] Computer Teaching and Research Section, Capital University of Physical Education and Sports, Beijing, China
[4] Faculty of Architecture, Computing and Humanities, University of Greenwich, London, UK

Corresponding author:
Hao Wu, College of Information Science and Technology, Beijing Normal University, Beijing 100875, China.
Email: wuhao@bnu.edu.cn

Sensors in sports can be attached to the user or an integral part of the (smart) sports equipment. Systems and applications in sports that are using wearable sensor data can be designed for a great variety of uses, from monitoring particular movements of an individual to overseeing the complete action in a group sports match.

According to the intended use, the complexity of the design varies considerably. Our plans are to develop biofeedback applications, particularly in biomechanical feedback systems, that would use sensors' data for providing the concurrent feedback to the user.[1] According to Sigrist et al.,[2] proper motor learning can be accelerated by the identification and prevention (interruption) of incorrectly performed actions. Our aim is to design and implement a real-time system that would notify the user about the incorrect action during the action itself or immediately after each period of a periodic activity.

As one of the state-of-the-art image classification approaches, convolutional neural network (CNN) aims to label elements with predefined classification tags on the basis of their resemblance; its striking success has aroused a surge of attention in computer vision, pattern recognition, and data mining owing to its automatic feature extraction, high accuracy, and high scalability in image classification, object detection, image retrieval, and image inpainting.[3] Due to the tremendous development in model incarnations, such as GoogLeNet[4] and ResNet,[5] and its high reliability and effectiveness in image classification,[6] we intend to transfer common CNN models into golf swing data classification, to improve the classification accuracy conducted by marking data with predefined labels from combinations of shapes and golf players. Encouraged by the impressive performance of CNN-based models on computer vision, CNN-based classifiers presented here are customized for one-dimensional (1D) golf swing signal classification. The candidate models take as input a golf swing datum composed of $n_s \times n_l$ data samples, where n_s denotes the number of channels and n_l denotes the length of sequences, and output the likelihood of which golf player and which swing it belongs to. The evaluation of accuracy on multiple combinations of sensors can basically illustrate the relevance of sensors attached to the smart golf club[7] and it is also a reference for reducing the input dimensionality.

The major contributions of this article are as follows:

- Four different state-of-the-art CNN-based classifiers are employed to classify 1D sequences of golf swing signals, group them, and mark them with the labels conducted by the combinations of golf players and shapes; meanwhile, it has been demonstrated that CNN-based models sufficiently show their dramatical superiority over the support vector machine (SVM) on behalf of the traditional methods.
- It is a beneficial trial of the well-devised multisensor selection to select the proper sensor or sensor combination in golf swing classification, which demonstrates the sensibility and reliability of these sensors.
- The evaluation on the real-world test set reveals the comparison and superiority of presented methods. Compared with the traditional CNN model, these complex models can be empowered to classify golf swings accurately. However, the traditional CNN model is still a feasible solution to golf classification due to the less calculation consumption.

This article is organized as follows: section "Related work" presents some related work concerning CNN and golf swing signal analysis. Section "Data collection" briefly introduces the smart golf club we used to collect golf swing data. Section "Methodology" describes the network model we design and some implementation details. Section "Experiments and results" presents the experiment we design and some experimental results to validate the effectiveness of our model. Section "Conclusion" concludes this article and lists some future work we plan to do.

Related work

The surge of devices with one or more sensors related to the smart sports equipment has been evident in recent years, and wearable devices used for professional sports detection and identification were widely used.[8–14] Some lightweight and small-size sensors were integrated into the smart sports equipment to measure and collect detailed physical and physiological quantities of sports activities including heart rate, accelerations, and others, and the collection of measurements and quantities associated with biological or physiological feedback has been proven beneficial to athlete performance improvement. The swing collection of golf wearable devices focused on motion sensors: accelerometers, gyroscopes, and magnetometers,[15] which are well-known as inertial measurement units (IMUs).[10–13] In our case, strain gage sensors, three-axis accelerometer, and three-axis gyroscope are leveraged to collect the motion and trajectory of golf players and golf balls.

Feature extraction methods extract a set of representative attributes from original datasets or data sequences, and these attributes can effectively capture latent properties that can be used to identify a distinct data record; furthermore, these attributes are defined in a lower dimension space rather than a high dimension space where original data records are defined, which can result in lower storage consumption and faster similarity calculation. Some well-known methods, such as ReliefF,[16] Laplacian score,[17] and Fisher score,[18] were common feature selection methods that were used to extract features from sports sequence data; they concentrated on and outputted features holding a high degree of correlation. With an appropriate manual configuration, they can achieve a promising performance in feature selection for sports sequence data.

The CNN (ConvNet) applications in computer vision have been achieving a striking success in recent years, especially in image classification, object detection, and image retrieve.[3,19] AlexNet[20] achieved a successful and groundbreaking performance in ImageNet Large Scale Visual Recognition Challenge (ILSVRC) competitions[21] in 2012, which brought into solving image classification a surge of development regarding ConvNet. A series of representative successors based on CNN architecture, such as VGGNet,[22] GoogLeNet,[4] and ResNet,[5] have gradually made progress one by one and reached better performance in terms of top-5 accuracy in image classification, even exceeded the human-level performance of 5.1% top-5 test error.[6] AlexNet[20] proposed ReLU activation function, local response normalization, and dropout, aiming to alleviate gradient vanishing/exploding, local data standardization, and alleviate overfitting, respectively. VGGNet[22] exploited a stack of smaller receptive fields to fit a bigger receptive field. GoogLeNet[4] built hierarchical Inception modules to extract multi-scale feature maps to understand deeper semantic context. ResNet[5] built skip connections within convolutional layers to propagate vanishing gradients backward. The trend of unsupervised learning based on the convolutional network has been raised since Generative Adversarial Network (GAN)[23] was proposed in which the network can be trained on its own with its specified discriminator and generator structure instead of the supervision of human being. Models with surprisingly low top-5 errors have been gradually proposed on the basis of reorganization and reconnection of intermediate layers, including ResNeXt[24] building parallel pathways and width-increasing shortcut connections within convolutional layers, FractalNet[25] leveraging fractal architecture to reorganize feature maps, and DenseNet[26] with dense blocks; thus, these novel architectures improved the statistics of classification quality of ImageNet competition.

CNN-based models have been extended to 1D classification,[27] such as automatic speech recognition,[28] electrocardiogram signal classification,[29,30] and biomedical time series classification.[31] CNN-based models are able to extract and leverage latent feature representations in time series with high tolerance of time translation; thus, results outperform methods based on hand-crafted features. We follow these 1D CNN classification implementations, formalize our golf classification, customize our CNN-based classifiers, and conduct experiments to validate their effectiveness among concurrent expressive classifiers.

Data collection

The sensor configuration of golf club used for collecting data is (a) two single-grid strain gage sensors detecting the golf club shaft bend, (b) one three-axis micro-electro-mechanical sensor (MEMS) accelerometer detecting acceleration, and (c) one three-axis MEMS gyroscope detecting the angular speed of the golf club. These four sensors sample a golf swing with the frequency of up to 500 Hz in 2 s and generate an eight-channel golf swing sample, each of which contains 1000 sampling points. Four professional and amateur golf players are selected to attempt to perform typical golf swings that are distinguishable and easy to label in accordance with prior professional golf swing knowledge. For the visualization of collecting golf swings, Figure 1 shows how a golfer performs a specific golf swing. Eventually, 213 typical golf swings were collected, presented in Table 1 and in Figure 2.

Methodology

In this section, we present the procedure of data preprocessing, CNN-based model design, and implementation details. We first introduce our data preprocessing involving data augmentation, data shuffling, and data standardization in sections "Data augmentation and shuffle" and "Data standardization." In addition, based on general CNN architecture, we present the customized vanilla CNN classifier and three advanced CNN classifiers derived from VGGNet,[22] GoogLeNet,[4] and ResNet[5] in section "Network architecture." Finally, we give some implementation details in section "Implementation details."

Data augmentation and shuffle

The function $x^{(i)}(t)$ is referred to as the representation of producing procedure of a golf swing with regard to time variable t and sensor i, so the sequence $y^{(i)}$ produced by sensor i can be denoted by equation (1)

$$y^{(i)} = x^{(i)}(t) \tag{1}$$

An example of the golf swing in our real-world dataset is shown in Figure 2. Each golf swing contains eight sequences from the strain gage sensors, the accelerometer, and the gyroscope sensor, where each sequence contains 1000 samples that are defined as Analysis Window and has been proved as the most significant part in the biofeedback system.[15] The analysis on the

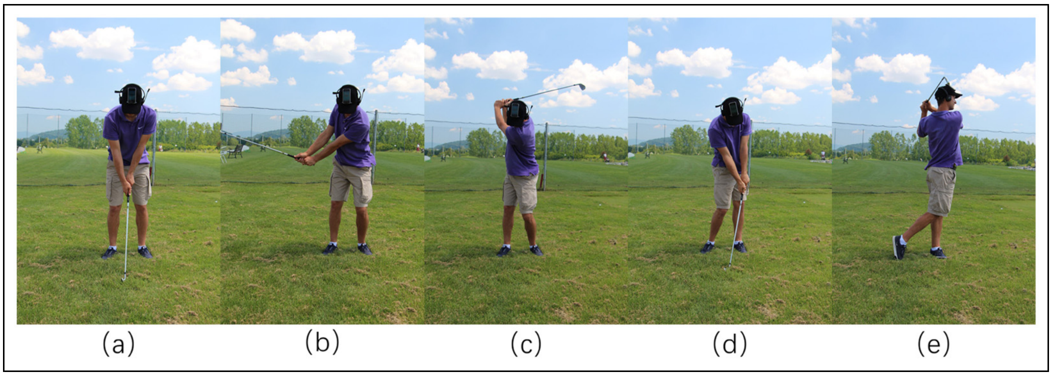


Figure 1. Collecting a specific golf swing. For the visualization of collecting golf swings, we present a series of photographs of a golfer depicting actions, which illustrate the routine of actions and the trajectory of the smart golf club. (a) Address—approximately before signal sample 400 in Figure 2. (b) backswing—approximately between signal sample 400 and 850 in Figure 2. (c) backswing—approximately at signal sample 850 in Figure 2. (d) Impact—at sample 999 in Figure 2. (e) Follow through—after sample 999 in Figure 2 (not shown in the signal). The attached sensors record the specific movements in a $n_s \times n_l$ tensor.

dataset containing the analysis window is valuable on the basis of the conclusion from Umek et al.[15]

For the robustness and balance among the dataset, we customize data augmentation strategy before the dataset is fed to our classifiers, including data scale-up, time left-translation, and time right-translation defined in equations (2) and (3), where the notations α and γ denote the rescaling factors, and the notation Δt denotes the time-translation factor

$$y^{(i)} = \alpha x^{(i)}(t \pm \Delta t) \tag{2}$$

$$y^{(i)} = \gamma x^{(i)}(t \pm \Delta t) \tag{3}$$

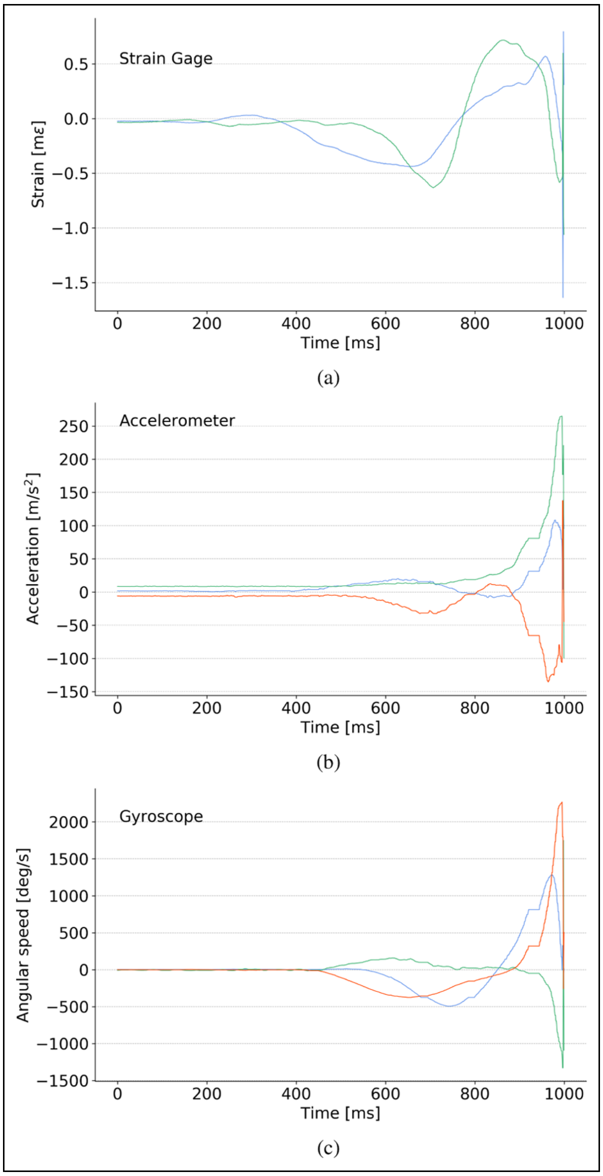
In the discussion of repeatability from Umek et al.,[15] sensor signals show high repeatability in both time and amplitude during several consistent swings from a specific player; that is, the deviation among swings of one motion from a player is very small. Consequently, the distortion of signals brought by equations (2) and (3) adequately empowers our classifier to robustly characterize golf swings. In practice, α, γ, and the sample delay Δt are set at 1.1, 0.9, and 3, respectively. Theoretically, the rescaling factors α and γ and the time left-translation factor Δt are hyperparameters and should be investigated in the experiment to evaluate the performance of our model. However, in practice, Umek et al.[15] has explored the repeatability of the smart golf club and have demonstrated that there exists little deviation in the real-world usage. Therefore, the default configurations of these factors are feasible to cover the deviation in the sampled data sequences and can guarantee the convincing experimental results.

The dataset is further augmented by oversampling the minorities of golf swings, which enlarges the minor golf swing set and balances the count of each class of golf swings. Minorities of golf swings are duplicated and transformed with the time-translation or data-rescaling equations defined in equation (2) or equation (3) randomly, and the whole dataset is shuffled randomly after augmentation as the augmented and shuffled training and test set are able to reinforce the robustness and evaluate the intrinsic feasibility of a well-trained classifier. Figure 3 shows an example of golf swing replicated by equations (2) and (3).

Eventually, the dataset contains 917 golf swing data composed of eight channels of 1000 samples, which is shown in Table 2 where ID, Golfer, Shape, Count, and AugCount denote the assigned numerical ID, the ID number of golf player generating these golf swings, the intended shape of these golf swings, the count of raw golf swings, and the count of the golf swing set augmented afterwards, respectively. Each swing is specifically labeled by the golf player ID and the intended shape and assigned with a unique numerical ID as a classification label. For training an effective model and evaluating it, the whole dataset is generally split into a training set containing two-third swings and a test set containing one-third swings without overlapping; the training set is in charge of training classifiers, while the test set evaluating them.

Data standardization

For scaling data to eliminate measured divergences in different sequences, data normalization is a widely used strategy to preprocess dataset to remove measures of sequences. Here, Z-score normalization is used to standardize each sequence by removing means and scaling to unit variance before classification in equation (4)



(a)

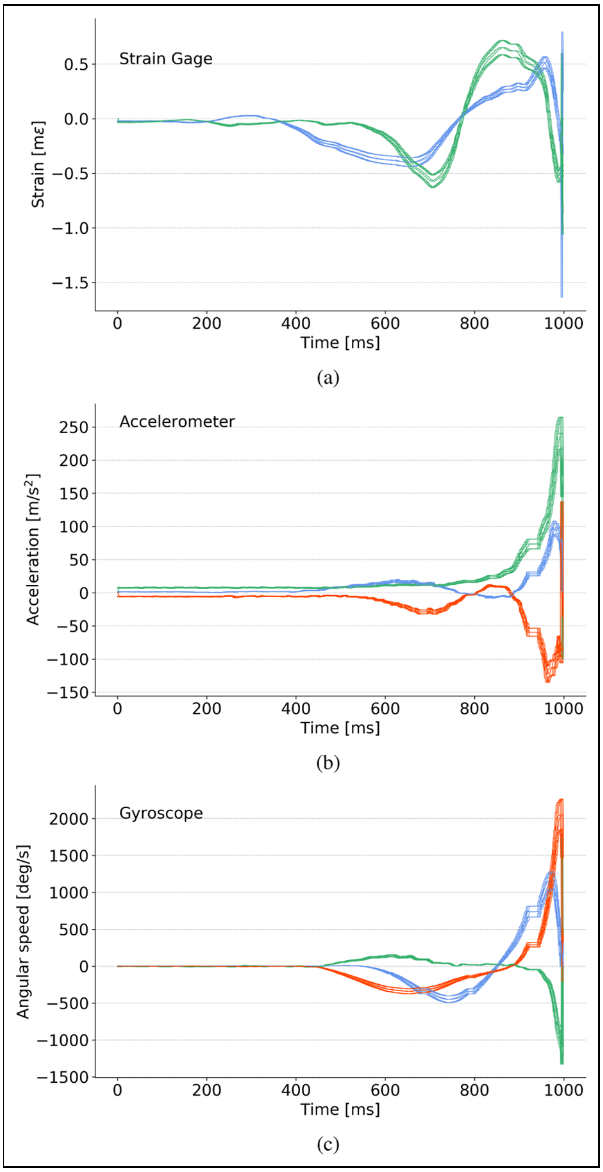
(b)

(c)

Figure 2. An example of a golf swing. Here, a golf swing is composed of eight-channel sequences from three kinds of sensors: (a) two sequences from SG sensors, (b) three from accelerometer, (c) and three from gyroscope.

$$y^{(i)'} = \frac{y^{(i)} - E\left[y^{(i)}\right]}{\sigma\left(y^{(i)}\right)} \tag{4}$$

Since some non-standardized strong sequences may manipulate the classification estimator, which may prevent the estimator learning from other sequences to calibrate parameters as expected, the data standardization[32,33] is employed to alleviate the domination of sequences from one individual sensor in training phase. In practice, some classification estimators based on gradient descent algorithm should take as input the standardized dataset since the convergence of estimators can be accelerated by leveraging the standardized dataset.



(a)

(b)

(c)

Figure 3. An example of data augmentation including data translating and data rescaling. We translate and rescale the golf swings in order to enrich our training and test set, and the translation and the rescaling functions are presented in equations (2) and (3): (a) sequences from SG sensors, (b) sequences from accelerometer, and (c) sequences from gyroscope.

An example of a standardized golf swing is shown in Figure 4. The mean of each channel is removed; the amplitude of each channel is rescaled as well, which results in eliminating the predominance of the single individual channel and enabling classification estimator to equally extract and learn features from each channel. In data processing, it is conventionally assumed that the training set and the test set are produced from a unique probability distribution; consequently, they share the same mean and standard deviation. So, the mean and standard deviation are calculated on the

Table 1. Meta-information about real-world golf swing dataset.

ID	Golfer	Shape	Count
0	2	Straight	30
1	1	Straight	79
2	4	Fade	2
3	1	Pull	9
4	4	Push-slice	1
5	2	Pull	1
6	1	Pull-hook	3
7	1	Slice	9
8	1	Push	9
9	4	Draw	4
10	1	Hook	5
11	4	Straight	11
12	3	Straight	15
13	4	Slice	2
14	1	Fade	7
15	1	Push-slice	1
16	4	Push	10
17	1	Draw	11
18	2	Push	4
Total count			213

Table 2. Meta-information about augmented golf swing data.

ID	Golfer	Shape	Count	AugCount
0	2	Straight	30	60
1	1	Straight	79	79
2	4	Fade	2	38
3	1	Pull	9	54
4	4	Push-slice	1	32
5	2	Pull	1	32
6	1	Pull-hook	3	42
7	1	Slice	9	54
8	1	Push	9	54
9	4	Draw	4	44
10	1	Hook	5	45
11	4	Straight	11	55
12	3	Straight	15	60
13	4	Slice	2	38
14	1	Fade	7	49
15	1	Push-slice	1	32
16	4	Push	10	50
17	1	Draw	11	55
18	2	Push	4	44
Total count			213	917

training set, and the data standardization is performed on both the training set and the test set with the above-mentioned mean and standard deviation.

Network architecture

Vanilla CNN. Here, we present our vanilla CNN architecture for golf swing classification, which conventionally contains three categories of layers[20,34,35] to extract features and classify convolutional layers, pooling layers, and fully connected layers, as Figure 5(a) shows. Vanilla CNN architecture distills and reorganizes representations latently distributed in golf swings by hierarchically stacking convolutional layers with learnable filters and pooling layers with a fixed stride layer by layer. Intermediate 1D convolutional layers automatically enable the representative relevant features, expand the receptive fields to gather more, and propagate them to the backward layers. Max-pooling layers reduce and constrain the dimensionality of feature maps by downsampling the coming feature maps and finally enhance the significant features. The backward fully connected layers classify those features and give likelihoods of golf swings; that is, they play a role of the classifier. The input golf swing signals contain eight channels and the output probability distributions indicate the posterior probability of classes of golf swings. Figure 5(a) shows the whole architecture of the vanilla CNN.

Table 3. Overall accuracy of CNN-based classifiers for different sensor types and their combinations in percent.

Model	All sensor (%)	SG (%)	Acc (%)	Gyro (%)
GolfVanillaCNN	95.04	96.03	97.68	66.67
GolfVGG	96.70	94.06	95.05	65.68
GolfInception	97.36	95.38	96.04	67.66
GolfResNet	92.07	96.70	95.71	66.67

CNN: convolutional neural network; SG: strain-gage sensors; Acc: single accelerometer sensor; Gyro: signal gyroscope sensor.

Here, we refer to ReLU[36,37] as the activation function. ReLU can fully transfer positive activation value to the following layers without any decay and eliminates negative activation value generated in the linear part of the current neuron. Furthermore, the gradients in backpropagation will not decay when passed by ReLU activation neuron; namely, ReLU is effectively able to alleviate the diffusion of gradients.[38]

The last layer consists of softmax[39] activation neurons, which outputs vectors representing the posterior probability of the input signals; the number of softmax neurons is given corresponding to the predefined categories of shapes and golf players.

VGG-like CNN. VGGNet[22] has, respectively, achieved a first and a second place in ILSVRC competition 2014 in localization and classification tracks, which brings a striking attention from tremendous researchers. The basis of VGGNet is a novel architecture that used very small receptive fields (3 × 3) in each convolutional stack composed of multiple convolutional layers, which made it possible to enlarge the depth to 16 or 19 layers due to the reduction of parameters and fully connected layers. The innovative improvement reveals that this configuration of deeper convolutional stacks with small

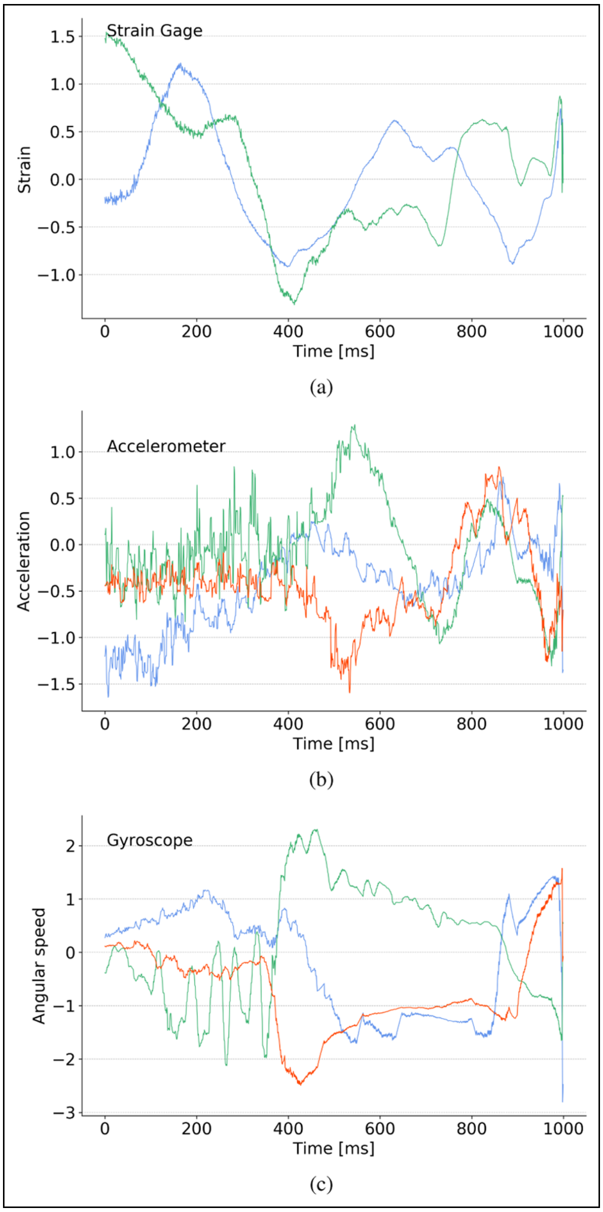


Figure 4. An example of data standardization. We standardize our golf swing dataset by removing the mean of the dataset and rescale it to unit variance, in order to amplify latent mere features: (a) standardized sequences from SG sensors, (b) standardized sequences from accelerometer, and (c) standardized sequences from gyroscope.

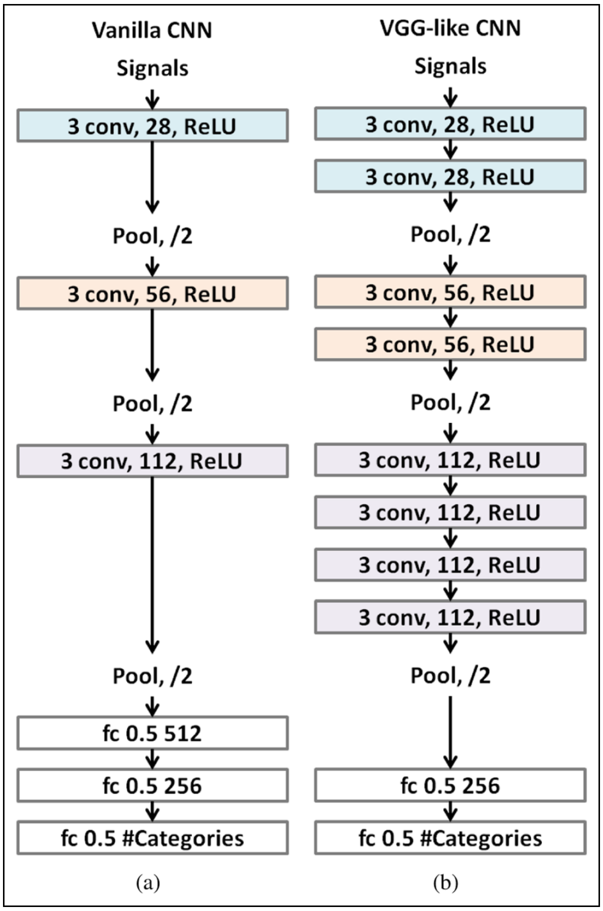


Figure 5. (a) Vanilla convolutional neural network for golf classification[27] and (b) VGG-like convolutional neural network for golf classification. We here use a vanilla CNN structure to evaluate the performance of CNN classifying golf swings, and we also build a VGG-like CNN classifier to test further. The VGG-like CNN classifier has wider receptive fields but uses fewer parameters in convolutional layers, which guarantees to extract the same scale features when reducing calculation consumption.

receptive fields can guarantee multiple feature extraction and facilitate further deep visual representations, which can lead to a great breakthrough in computer vision.

We also reinforce our golf feature extraction by stacking convolutional layers with small receptive fields hierarchically and remove a fully connected layer with 512 activation units to reduce the number of parameters. The first and second convolutional layers are substituted with stacks of convolutional layers composed of two convolutional layers with small kernels of size 3, while the third convolutional layer is substituted with a stack composed of four convolutional layers with small kernels of size 3. Max-pooling layers are maintained to reduce the size of feature maps, and two fully connected layers follow the stacks of convolutional layers and max-pooling layers to classify non-linearly. The whole architecture is shown in Figure 5(b).

Inception-based CNN. In spite of achieving a striking success with small receptive fields, such as VGG net, a multi-scale comprehension conducted by multi-scale receptive fields can be applied to ameliorate the accuracy of image classification, which increases the depth

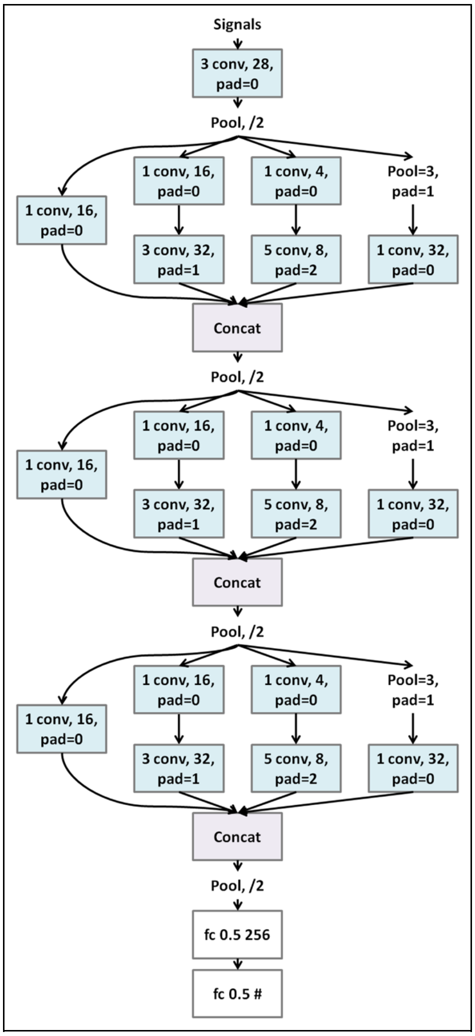


Figure 6. Convolutional neural network for golf classification with Inception module. Inception module concatenates multi-scale feature maps convolved by prior convolutional layers, aiming to boot performance by extracting multi-scale features within this module.

and width of the network. In computer vision, Inception module is a concatenation of multi-scale receptive fields, from 1×1, 3×3, and 5×5, and a downsampling component implemented by pooling, which performs a multi-scale feature extraction after concatenating. In practice, receptive fields of size 1×1 in Inception module can reduce the dimensionality of feature maps to avoid the dimensionality exploding, all of which can decay the growing computational budget caused by increasing depth and width.

Our Inception-based network hierarchically aggregates three 1D Inception module, where the sizes of the filter kernels are 1, 3, and 5, respectively. Inner convolutional layers with 1-sized filters facilitate dimensionality reduction since the number of feature maps would not explode after multiple convolutional layers. Pooling layers are preserved, while a fully connected layer is also removed in practice, which is of use in reducing the dimensionality of feature maps, decreasing the number of parameters, and declining the computational budgets. The architecture is shown in Figure 6.

Residual-block-based CNN. In practice, the CNN is more and more difficult to train when the layers are accumulated deeper and deeper since the gradients inevitably vanish or explode when propagated backward. He et al.[5] proposed a residual learning module to address the gradient vanishing/exploding issue, which built shortcut connections in residual blocks to pass the identity map x sideways. Hybrid gradients can be decomposed into one term connecting to intermediate layers and another term skipping the intermediate layers, which can guarantee that the gradient can be directly propagated to shallower layers to reinforce the calibration.[40] The residual block can be described in equation (5)

$$x^{(\ell)} = \mathcal{F}\left(x^{(\ell-1)}\right) + x^{(\ell-1)} \qquad (5)$$

Inspired by He et al.,[5] we customize our model by superseding intermediate convolutional layers with concatenations of a residual block stack and a dimensionality-increasing residual block. The residual block takes as input the feature maps $x^{(\ell-1)}$ from the previous layer, propagates them forward through a size-maintained stack of convolutional layers, and meanwhile outputs the element-wise sum of filtered feature maps and identity maps $x^{(\ell-1)}$. The dimensionality-increasing residual block similarly convolves and passes the input feature maps and identity maps sideways as an ordinary residual block does, but the number of feature maps is increased in the intermediate convolutional layers. Three convolutional layers are superseded by residual block stacks and dimensionality-increasing residual blocks, whereas a fully connected layer is removed for the purpose of reducing the number of parameters. Our residual-block-based CNN is shown in Figure 7.

Implementation details

The architecture of these four CNN-based classifiers are shown in Figures 5, 6, and 7. The vanilla CNN classifier is composed of three convolutional layers, three max-pooling layers, and three fully connected layers. Layers 1, 3, and 5 are convolutional layers that formulated with 28, 56, and 112 trainable kernels, respectively. The last three fully connected layers contain 512, 259, and 19 neurons with 0.5 dropout to alleviate the overfitting. ReLU is imposed in the intermediate layers and softmax neurons in the last layer of classification, as is shown in Figure 5(a).

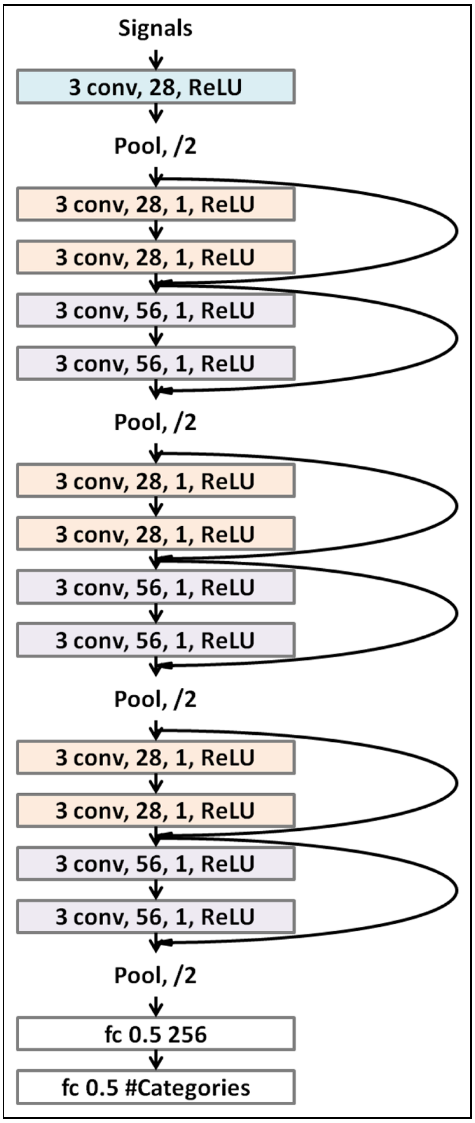


Figure 7. Convolutional neural network for golf classification with residual block module. Residual blocks pass gradients through the identity map to calibrate and converge faster when propagating gradients backward.

As for the VGG-like CNN, the stacked convolutional layers with small receptive fields replace the original single-layer convolutional layers. The three stacks contain 2, 2, and 4 convolutional layers with ReLU activation neurons, and a fully connected layer with 512 neurons is removed in the implementation; the reason why a fully connected layer is removed is that the nonlinear classifier part should be weakened if the feature extraction part has been enhanced. Other parameters are the same as the vanilla CNN, as shown in Figure 5(b).

The Inception-based CNN stacks Inception modules in which the convolutional layers with multi-scale filters are concatenated after the first convolutional layer. The convolutional layers with 1-size filters are employed in Inception modules in order to avoid the dimensionality explosion. The fully connected layers and parameters are the same as the VGG-like CNN, as shown in Figure 6.

The residual-block-based CNN stacks residual blocks where shortcut connections are built to skip two convolutional layers to pass the residual errors. In a residual block, the first two convolutional layers do not expand the volumes of feature maps, whereas the last two convolutional layers double the volumes of feature maps. The fully connected layers and parameters are the same as the VGG-like CNN, as shown in Figure 7.

The input dimensionality of CNN-based models is $n_{batch} \times n_s \times n_l$, where n_{batch} denotes the size of a mini-batch of golf swings, n_s denotes the number of channels, and n_l denotes the sequence lengths of a golf swing. Since networks perform classification task and the output vectors represent the posterior probability distribution, the loss function category cross entropy should no doubt be employed here, which is defined by equation (6)

$$\bar{L} = -\frac{1}{|B|}\sum_{i\in B}\sum_{j} t_{i,j}\log(o_{i,j}) \qquad (6)$$

The optimizer ADAM[41] minimizes the loss function of the defined category cross entropy and offers the gradients for the global trainable variables for the convergence of the loss function.[42]

Deterministic CNN models output the posterior probability of input golf swings, and the prediction is determined by the maximum of the posterior probability that is calculated in equation (7)

$$l_i = \arg\max_{j}(o_{i,j}) \qquad (7)$$

Experiments and results

Experimental dataset

First, we review our real-world golf swing dataset briefly. Our device sampled 213 golf swings produced by four professional or amateur golf players marked by aliases 1–4, and the dataset contains nine distinct golf swing shapes at the most from one single golf player; there are totally 19 categories of different labels of combined shapes and golf players in the dataset. The dataset has been balanced, augmented, shuffled, and standardized by the strategy in sections “Data augmentation and shuffle” and “Data standardization” and has been presented in Tables 1 and 2.

The balance and randomness of the multi-class dataset have been guaranteed by the strategies of data

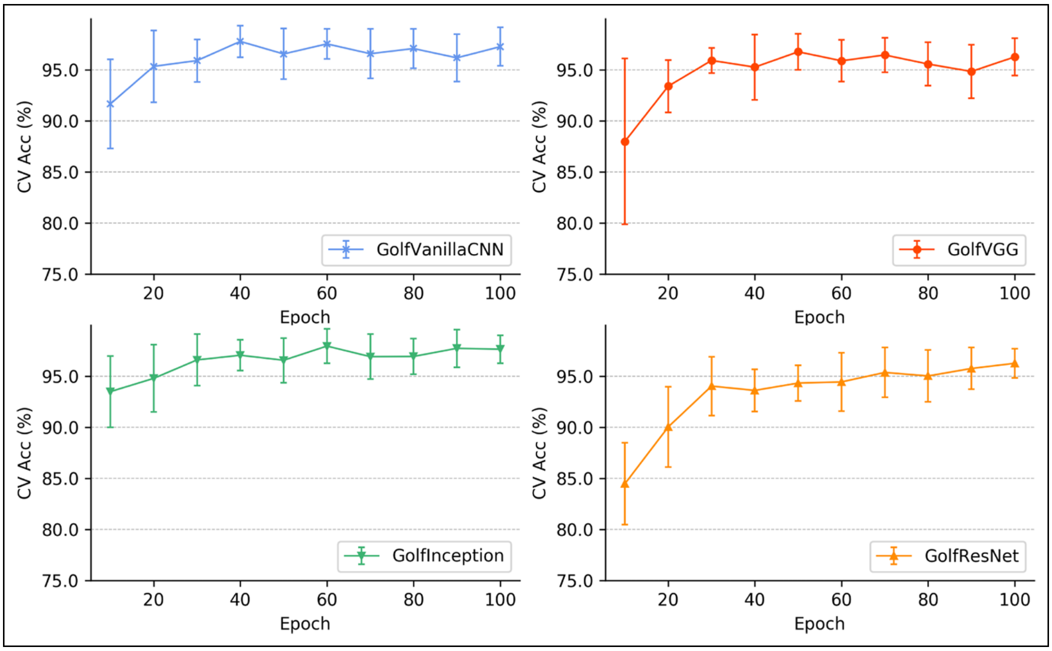


Figure 8. 10-fold cross-validation results of four different models. We can find that these four classifiers converge and achieve feasible accuracy, which proves that these four classifiers are adequate to classify golf swings accurately on validation sets. The overfitting evaluation is discussed in sections "Hyperparameter selection", "Overall accuracy," and "Precision-Recall Evaluation."

augmentation and data shuffling mentioned in sections "Data augmentation and shuffle" and "Data standardization." The minorities of golf swings have been dramatically enlarged but the counts of majorities maintained; the difference of count still exists at the same time, so it is believed that the strategies guarantee the inherent distribution of the original real-world golf swing dataset, while the oversampling, random disturbance, and shuffling bring the interferences into the augmented dataset to reinforce and test the robustness of CNN-based classifiers.

The augmented dataset is separated into a training set containing two-third of swings and a test set containing the remaining one-third of data. The 10-fold cross-validation is performed on the training set to select proper hyperparameters and models. Classifiers are implemented by Theano[43] and Lasagne[44] trained on the NVIDIA® CUDA® accelerators. So, in the next parts, we can discuss evaluation indicators including the overall accuracy, precision–recall indicators and curves, and F1 scores.

Hyperparameter selection

The 10-fold cross-validation is employed to select the hyperparameters dominating the accuracy of golf swing classification. Since the architecture of CNN-based classification model has been fixed and the learning rate and the optimizer have been settled empirically, 10-fold cross-validation can be used to select the models, sensor combination, and the number of epochs. With other hyperparameters fixed, the candidate hyperparameters should be selected from a candidate range or set that is predefined rationally and empirically according to the prior knowledge.[7] The selections of models and sensor combinations are, respectively, given in sections "Experimental dataset" and "Hyperparameter selection"; meanwhile, the conclusion of epoch selection can be summarized from these two sections.

Model selection. The four common CNN-based models including vanilla CNN model implemented as GolfVanillaCNN, VGG-like model implemented as GolfVGG, Inception-based model implemented as GolfInception, and residual-block-based model implemented as GolfResNet are evaluated here with all sensors and longest sequence length selected in order to figure out in terms of accuracy which model can classify more accurately than other models. These four CNN-based models are constructed and trained on 10 training subsets and evaluated 10 validation subsets from 10-fold cross-validation. The indicators involving means of accuracy and standard deviations show the performance of these models in classification accuracy that is adequately solid to demonstrate the effectiveness of the models. The means of accuracy and standard deviations with respect to epochs and models are shown in Figure 8.

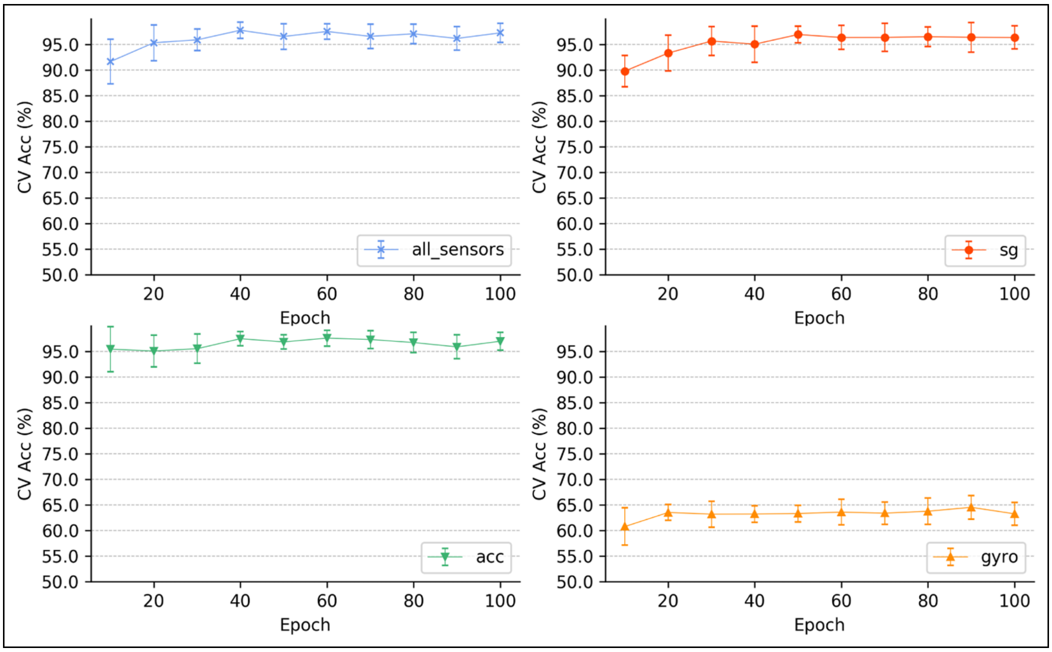


Figure 9. 10-fold cross-validation for four different sensors (or joint sensor combination). We validate sequences sampled from SG sensors, accelerometer, gyroscope, and joint sensor combination (all sensors) and find that sequences from all sensors, SG sensors, and accelerometer hold enough features to be grouped into proper classes except gyroscope, so these three kinds of sensor (or combination) are used to represent golf swings.

From Figure 8, it can be concluded that these four models are adequate to classify golf swing data accurately, and the selection of models has less effect on the classification performance in terms of accuracy. It is obvious that the final means of accuracy from these four models are close enough and almost reach 97.5%, which basically demonstrates the effectiveness of these four models in classification. In addition, the less discrepancy of accuracy and narrower standard deviations among these four models with epochs increasing can sufficiently support that the selection of models matters less finally, namely, the vanilla CNN can easily group the golf swing data. In this case, either of these four models can be selected to classify the test set of golf swing data.

Sensor selection. The exploration of sensor selection is meaningful since the dimensionality of the input golf swing can be abundantly reduced as long as signals from single one sensor carry sufficient information for classification; furthermore, the time consumption can drop dramatically, which can be a guarantee of real-time analysis. Signals from all three sensors, together with the single-strain-gage sensors (sg), the single accelerometer sensor (acc), and the signal gyroscope sensor (gyro), are fed into vanilla CNN model to test which sensor (or combination) is the most sensitive when classifying. Figure 9 shows the cross-validation result of sensor (combination) selection.

From Figure 9, it is obviously concluded that all sensors, the sg sensors, and the accelerometer are effective in golf swing classification; on the contrary, gyroscope fails in classification on the whole. The convergence of means of accuracy among all sensors, the sg sensors, and the accelerometer shows that it is believed that these sensors or combination may be identical in classification, which provides the practical basis for dimensionality reduction. However, signals from the vulnerable gyroscope cannot enable vanilla CNN model to classify accurately, and the coherency of low means of accuracy emphasizes its invalidity, so gyroscope could perhaps be one of the candidate sensors that can be eliminated to speed up the classification. In this case, all sensors, the sg sensors, and the accelerometer are able to be used to brew a well-trained accurate model except the vulnerable gyroscope.

Epoch selection. The selection of epochs is tested in two aforementioned cross-validations of hyperparameter selections. It is concluded that the performance of CNN-based classifiers can be improved gradually with the increasing epochs in terms of accuracy. The means of accuracy can be improved and the standard deviations of accuracy can be narrowed apparently, and the

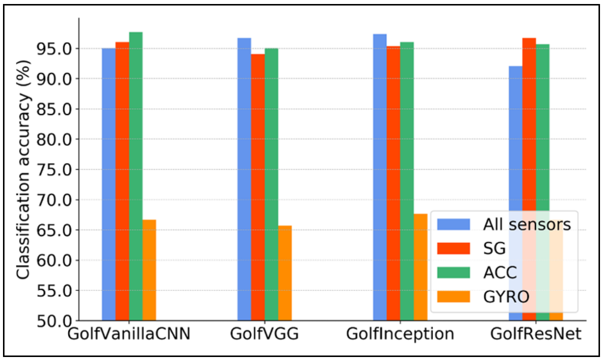


Figure 10. Test accuracy of CNN-based classifiers. We test four above-mentioned CNN-based classifiers and SVM classifier to evaluate the generalization and the superiority of CNN-based classifiers. In this figure, it is concluded that the consistency of superb performance over SVM classifier demonstrate that CNN-based classifiers generalize their acceptable performance to the test set and outperform SVM classifier on behalf of some traditional classification models.

convergence of means of accuracy and standard deviations supports the improvement of accuracy with respect to increasing epochs solidly. In this case, 100 epochs should be employed to train the CNN-based models iteratively.

Conclusions of hyperparameter selection. Some conclusions can be made in terms of the aforementioned hyperparameter selection.

- The advanced model architecture has less effect on the classification performance in terms of accuracy that reaches to 95%, which means all of them are acceptable in golf swing classification.
- All the sensor combination, the strain gage sensors, and the accelerometer are effective in golf swing classification, while the vulnerable gyroscope could fail on account of its unpredicted invalidity; it is positive that CNN-based model can label data properly with even one single sensor.
- Sufficient epochs are the guarantee of higher validation accuracy and less standard deviation, namely, stable classifiers are brewed after 100 iterations of training.

Overall accuracy

Accuracy should be first discussed since it is a principal and premier indicator evaluating the performance and effectiveness of classifiers. The four common CNN-based models are well-trained on the whole training set and tested on the preserved test set, fed with signals from multiple sensor selections, and eventually, they are evaluated in accordance with accuracy. The multiple evaluation results are shown and illustrated in Table 3 and Figure 10.

The comparison of accuracy first reveals the superiority of CNN-based classifiers. We attribute this to CNN-based classifiers extracting features more accurately. CNN-based classifiers extract translation-invariant features from data sequences automatically, classify them with the latter fully connected layers, and calibrate learnable receptive fields if they find errors between their outputs and references. These end-to-end models directly translate standardized data into corresponding labels without separated preprocessing, which exploits less prior knowledge distilled from original data and help to identify golf swings automatically.

We also find that the accuracy on the test set confirms the coherency of accuracy corresponding to variants of CNN models from 10-folds cross-validation: fed with selective sensors or combination, the classifiers based on advanced CNN components can group golf swing data from the preserved test set properly in terms of accuracy after sufficient epochs. Classifiers fed with signals from gyroscope perform inaccurately as it did in 10-folds cross-validation. Consequently, the superiority over SVM and the coherency of the accuracy of CNN-based models demonstrate the advance of the performance of CNN-based models in golf swing classification tasks.

Precision–recall evaluation

Precision–recall indicators and curves[45] are employed to show the superiority of CNN-based classifiers in comparison with SVM on behalf of traditional classifiers. In this case, precision is a measure reflecting golf swing relevancy, which indicates the rate of correct golf swings CNN-based classifiers retrieve; recall is a measure reflecting how many truly relevant golf swings are retrieved, which indicates the sensitivity of CNN-based classifiers when the classifiers are confronting plausibly incorrect golf swings. In this multi-class classification case, micro precision–recall indicators and curves are presented to evaluate SVM- and CNN-based classifiers, which are shown in Tables 4–8 and Figure 11.

We focus on the overfitting issue on each class. The desired golf classifier should identify all correct golf swings and distinguish errors accurately; therefore, both the precision and recall scores are 1.0. However, the generalization of practical classifiers is not adequate to classify data perfectly, so we focus on the tradeoff between precision and recall and hope CNN-based classifiers are sensitive to errors as well as keep precision. From, Tables 4–8, we can observe that CNN-based classifiers perform better than SVM in terms of averages of both precision scores and recall scores, which indicates that CNN-based model can group

Table 4. Precision–recall of SVM classifier.

	Precision	Recall	f1 score	support
0	1.00	0.62	0.77	24
1	0.93	0.90	0.91	29
2	1.00	1.00	1.00	11
3	1.00	1.00	1.00	17
4	1.00	1.00	1.00	9
5	1.00	1.00	1.00	11
6	0.61	1.00	0.76	14
7	0.94	0.89	0.92	19
8	0.93	0.81	0.87	16
9	0.85	0.73	0.79	15
10	1.00	1.00	1.00	16
11	0.56	0.88	0.68	17
12	1.00	1.00	1.00	19
13	1.00	0.33	0.50	12
14	0.88	0.88	0.88	16
15	1.00	1.00	1.00	10
16	0.87	0.87	0.87	15
17	1.00	0.78	0.88	23
18	0.53	1.00	0.69	10
Average/total	0.91	0.87	0.87	303

SVM: support vector machine.

Table 6. Precision–recall of GolfVGG classifier.

	Precision	Recall	f1 score	Support
0	1.00	0.92	0.96	24
1	1.00	0.79	0.88	29
2	1.00	1.00	1.00	11
3	0.94	1.00	0.97	17
4	1.00	1.00	1.00	9
5	1.00	1.00	1.00	11
6	1.00	1.00	1.00	14
7	1.00	1.00	1.00	19
8	1.00	0.94	0.97	16
9	1.00	1.00	1.00	15
10	0.94	1.00	0.97	16
11	0.94	1.00	0.97	17
12	0.86	1.00	0.93	19
13	1.00	1.00	1.00	12
14	1.00	0.94	0.97	16
15	1.00	1.00	1.00	10
16	0.94	1.00	0.97	15
17	1.00	1.00	1.00	23
18	0.77	1.00	0.87	10
Average/total	0.97	0.97	0.97	303

Table 5. Precision–recall of GolfVanilla CNN classifier.

	Precision	Recall	f1 score	Support
0	1.00	0.83	0.91	24
1	0.88	0.79	0.84	29
2	0.92	1.00	0.96	11
3	0.85	1.00	0.92	17
4	1.00	1.00	1.00	9
5	1.00	1.00	1.00	11
6	0.93	1.00	0.97	14
7	1.00	1.00	1.00	19
8	1.00	1.00	1.00	16
9	1.00	1.00	1.00	15
10	0.94	1.00	0.97	16
11	1.00	0.94	0.97	17
12	1.00	1.00	1.00	19
13	1.00	1.00	1.00	12
14	1.00	1.00	1.00	16
15	1.00	1.00	1.00	10
16	1.00	0.93	0.97	15
17	0.91	0.87	0.89	23
18	0.71	1.00	0.83	10
Average/total	0.95	0.95	0.95	303

CNN: convolutional neural network.

Table 7. Precision–recall of GolfInception classifier.

	Precision	Recall	f1 score	Support
0	0.96	0.92	0.94	24
1	1.00	0.79	0.88	29
2	1.00	1.00	1.00	11
3	0.94	1.00	0.97	17
4	1.00	1.00	1.00	9
5	1.00	1.00	1.00	11
6	1.00	1.00	1.00	14
7	1.00	1.00	1.00	19
8	0.94	1.00	0.97	16
9	1.00	1.00	1.00	15
10	0.94	1.00	0.97	16
11	1.00	1.00	1.00	17
12	1.00	1.00	1.00	19
13	1.00	1.00	1.00	12
14	1.00	1.00	1.00	16
15	1.00	1.00	1.00	10
16	1.00	1.00	1.00	15
17	0.92	1.00	0.96	23
18	0.83	1.00	0.91	10
Average/total	0.98	0.97	0.97	303

swings into proper classes while assigning proper labels. It is concluded that CNN-based classifiers are adequate to overcome overfitting on our test set, which can guarantee the usability of CNN-based classifiers in real-world datasets.

From Figure 11, it is found out that CNN-based classifiers perform superbly in comparison with the SVM classifier with whichever sensor is selected to use and the sequence length is fixed. Precision–recall curves from CNN-based classifiers are universally above the curve from SVM classifier, which means CNN-based classifiers quantitatively exceed SVM classifier in terms of precision and recall indicators. In addition, it is likely that the vulnerable gyroscope may be easy to be intervened and is disabled to produce distinguishable signals, and errors in Figure 11(d) reveal it as well.

Table 8. Precision–recall of GolfResNet classifier.

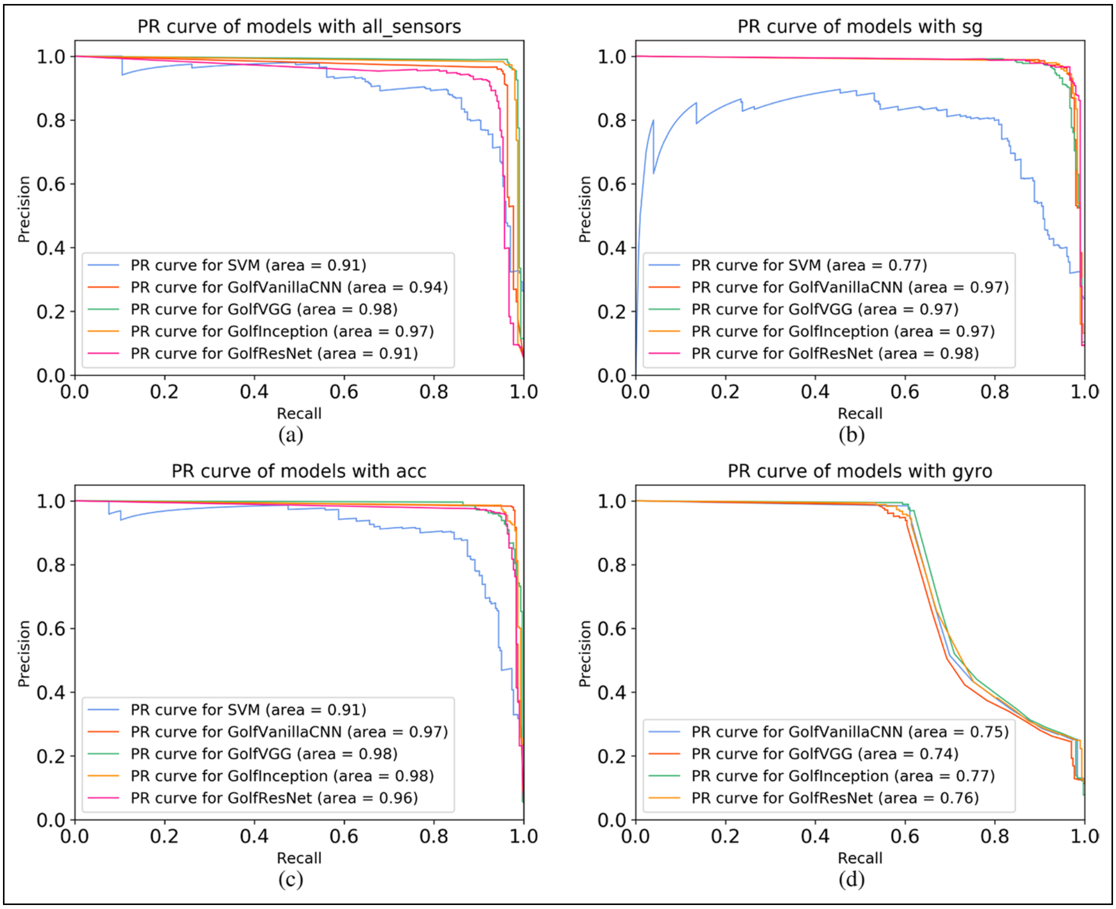
	Precision	Recall	f1 score	Support
0	1.00	0.83	0.91	24
1	1.00	0.76	0.86	29
2	0.92	1.00	0.96	11
3	0.77	1.00	0.87	17
4	1.00	1.00	1.00	9
5	1.00	1.00	1.00	11
6	1.00	1.00	1.00	14
7	1.00	0.95	0.97	19
8	0.67	1.00	0.80	16
9	1.00	1.00	1.00	15
10	1.00	0.88	0.93	16
11	0.84	0.94	0.89	17
12	1.00	1.00	1.00	19
13	1.00	0.92	0.96	12
14	1.00	0.69	0.81	16
15	1.00	1.00	1.00	10
16	0.92	0.80	0.86	15
17	0.92	1.00	0.96	23
18	0.71	1.00	0.83	10
Average/total	0.94	0.92	0.92	303

F1 score evaluation

F1 score[45] is a synthetic indicator taking into consideration both precision and recall; it is interpreted as a weighted average of precision and recall, as shown in equation (8) where P denotes precision and R recall

$$F1 = 2 \cdot \frac{P \times R}{P + R} \tag{8}$$

The F1 score takes into account both precision and recall, so it can indicate the performance of the classifier synthetically. In the multi-class evaluation, the micro F1 score takes into consideration all the true positives (TPs), false negatives (FNs), and false positives (FPs) and calculates the F1 score globally; on the contrary, the macro F1 score calculates the unweighted mean of all the F1 scores for each class. Although the micro F1 score takes label imbalance into account, these two scores indicate the classification performance equivalently in our cases, since the classes of golf swings have been balanced in the preprocessing. In our case, a classifier performs perfectly when its F1 score reaches 1.0



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

Figure 11. (a) Micro precision–recall Curves with respect to all sensors, (b) micro precision–recall curves with respect to strain gage sensors, (c) micro precision–recall curves with respect to accelerometer, and (d) micro precision–recall curves with respect to gyroscope.

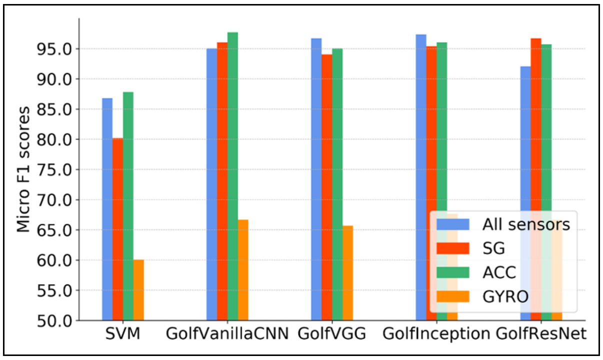


Figure 12. Micro F1 score with respect to all sensors, SG, accelerometer, and gyroscope.

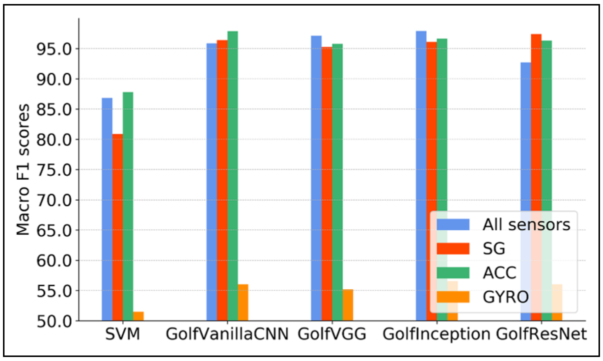


Figure 13. Macro F1 score with respect to all sensors, SG, accelerometer, and gyroscope.

since it retrieves all correct golf swings without missing ones and all retrieved golf swings are highly relevant. So, the comparison of the F1 score within CNN-based classifiers and SVM classifier presents the performance in golf swing classification. The experimental results are shown in Figures 12 and 13.

From Figures 12 and 13, it is found that CNN-based classifiers perform superbly in comparison with SVM classifier with whichever sensor and sequence length are selected to use. F1 scores of CNN-based classifiers are universally over scores of SVM in Figures 12 and 13, which means CNN-based classifiers quantitatively exceed SVM classifier when precision and recall are together taken into consideration; in addition to model selection, the strain gage sensors or the accelerometer can independently enable classifiers to classify correctly. At the same time, it is demonstrated again that the vulnerable gyroscope is easy to be intervened and may not produce the distinguishable signals.

Conclusion

In this article, we investigate golf swing data classification methods based on varieties of classifiers of deep CNNs fed with multi-sensor sequences. The CNN-based classifiers are adequate to correctly group the multi-channel golf swing data labeled by the hybrid categories from different golf players and shapes and quantitatively outperform SVM classifier in terms of widely accepted evaluation indicators including accuracy, precision–recall indicators and curves, and F1 scores on the preserved test set. Some conclusions are proclaimed again here.

- The indicators including accuracy, precision–recall curves, and F1 scores can quantitatively demonstrate that CNN-based classifiers can reach the acceptable accuracy in the golf swing classification tasks and outperform the SVM classifier.
- The consistent performance of accuracy among sensors can demonstrate that signals from even one single sensor can be adequate in identifying shapes of golf swings, while the vulnerable gyroscope is easy to be intervened and may not individually produce distinguishable signals.
- It has been illustrated that CNN-based classifiers are basically tolerant with the time translation and other plausibly existed noise imported initially since the consistency of indicators is observed in the 10-fold cross-validation and the test phase.

In future, we plan to investigate the default hyperparameter configuration α, γ, and Δt; the interior reason why gyroscope is invalid in our context; and how to decrease the probability of gyroscope invalidity. Furthermore, we explore the effectiveness of CNN-based classifiers on a larger real-world dataset and discover more evidence that can demonstrate the availability of CNN-based classifiers in a real-time or high-noise analysis context.

Declaration of conflicting interests

The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

Funding

The author(s) disclosed receipt of the following financial support for the research, authorship, and/or publication of this article: This research is partially sponsored by National Natural Science Foundation of China (Nos 61571049, 61401029, 11401028, 61472044, 61472403, and 61601033), the Fundamental Research Funds for the Central Universities (Nos 2014KJJCB32 and 2013NT57), China Postdoctoral Science Foundation–Funded Project (No. 2016M590337), the BNU Graduate Students' Platform for Innovation & Entrepreneurship Training Program (No. 3122121F1), the Slovenian Research Agency within the research program Algorithms and Optimization Methods in Telecommunications (research core funding No. P2-0246), and by SRF for ROCS and SEM.

References

1. Umek A, Tomažič S and Kos A. Wearable training system with real-time biofeedback and gesture user interface. *Pers Ubiquit Comput* 2015; 19(7): 989–998.
2. Sigrist R, Rauter G, Riener R, et al. Augmented visual, auditory, haptic, and multimodal feedback in motor learning: a review. *Psychon Bull Rev* 2013; 20(1): 21–53.
3. Guo Y, Liu Y, Oerlemans A, et al. Deep learning for visual understanding: a review. *Neurocomputing* 2016; 187: 27–48.
4. Szegedy C, Liu W, Jia Y, et al. Going deeper with convolutions. In: *The IEEE conference on computer vision and pattern recognition (CVPR)*, Boston, MA, 7–12 June 2015, pp.1–9. New York: IEEE.
5. He K, Zhang X, Ren S, et al. Deep residual learning for image recognition. In: *The IEEE conference on computer vision and pattern recognition (CVPR)*, Las Vegas, NV, 27–30 June 2016, pp.770–778. New York: IEEE.
6. He K, Zhang X, Ren S, et al. Delving deep into rectifiers: surpassing human-level performance on imagenet classification. In: *Proceedings of the IEEE international conference on computer vision*, Santiago, Chile, 7–13 December 2015, pp.1026–1034. New York: IEEE.
7. Umek A and Kos A. The role of high performance computing and communication for real-time biofeedback in sport. *Math Probl Eng* 2016; 2016: 4829452.
8. Lightman K. Silicon gets sporty. *IEEE Spectrum* 2016; 53(3): 48–53.
9. Akhras G. Smart materials and smart systems for the future. *Can Mil J* 2000; 1(3): 25–31.
10. Chambers R, Gabbett TJ, Cole MH, et al. The use of wearable microsensors to quantify sport-specific movements. *Sports Med* 2015; 45(7): 1065–1081.
11. Yu G, Jang YJ, Kim J, et al. Potential of IMU sensors in performance analysis of professional alpine skiers. *Sensors* 2016; 16(4): 463.
12. Hsu YL, Chen YT, Chou PH, et al. Golf swing motion detection using an inertial-sensor-based portable instrument. In: *IEEE international conference on consumer electronics-Taiwan (ICCE-TW)*, Nantou, Taiwan, 27–29 May 2016, pp.1–2. New York: IEEE.
13. Mendes JJA, Vieira MEM, Pires MB, et al. Sensor fusion and smart sensor in sports and biomedical applications. *Sensors* 2016; 16(10): 1569.
14. Abdul Razak AH, Zayegh A, Begg RK, et al. Foot plantar pressure measurement system: a review. *Sensors* 2012; 12(7): 9884–9912.
15. Umek A, Zhang Y, Tomažič S, et al. Suitability of strain gage sensors for integration into smart sport equipment: a golf club example. *Sensors* 2017; 17(4): E916.
16. Robnik-Šikonja M and Kononenko I. Theoretical and empirical analysis of ReliefF and RReliefF. *Mach Learn* 2003; 53(1–2): 23–69.
17. He X, Cai D and Niyogi P. Laplacian score for feature selection. In: *NIPS'05 proceedings of the 18th international conference on neural information processing systems*, Vancouver, BC, Canada, 5–8 December 2005, pp.507–514. Cambridge, MA: MIT Press.
18. Bishop CM. *Neural networks for pattern recognition*. Oxford, NY: Oxford university press, 1995.
19. Wu H, Li Y, Bi X, et al. Joint entropy based learning model for image retrieval. *J Vis Commun Image R* 2018; 55: 415–423.
20. Krizhevsky A, Sutskever I and Hinton GE. Imagenet classification with deep convolutional neural networks. In: *International conference on neural information processing systems*, 2012, pp.1097–1105. Curran Associates Inc.
21. Russakovsky O, Deng J, Su H, et al. ImageNet Large Scale Visual Recognition Challenge. *Int J Comput Vision* 2015; 115(3): 211–252.
22. Simonyan K and Zisserman A. Very deep convolutional networks for large-scale image recognition. arXiv6 preprint arXiv:14091556, 2014.
23. Goodfellow I, Pouget-Abadie J, Mirza M, et al. Generative adversarial nets. In: Ghahramani Z, Welling M, Cortes C, et al. (eds) *International conference on neural information processing systems*, 2014, pp.2672–2680. MIT Press.
24. Xie S, Girshick R, Dollár P, et al. Aggregated residual transformations for deep neural networks. arXiv preprint arXiv:161105431, 2016.
25. Larsson G, Maire M and Shakhnarovich G. Fractalnet: ultra-deep neural networks without residuals. arXiv preprint arXiv:160507648, 2016.
26. Huang G, Liu Z, Weinberger KQ, et al. Densely connected convolutional networks. arXiv preprint arXiv:160 80 6993, 2016.
27. Jiao L, Wu H, Bie R, et al. Multi-sensor golf swing classification using deep CNN. *Procedia Comput Sci* 2018; 129: 59–65.
28. Golik P, Tüske Z, Schlüter R, et al. Convolutional neural networks for acoustic modeling of raw time signal in LVCSR. In: *Sixteenth annual conference of the international speech communication association*, Dresden, 6–10 September 2015, pp.26–30. Red Hook, NY: Curran Associates.
29. Al Rahhal MM, Bazi Y, AlHichri H, et al. Deep learning approach for active classification of electrocardiogram signals. *Inform Sciences* 2016; 345: 340–354.
30. Jin LP and Dong J. Deep learning research on clinical electrocardiogram analysis. *Sci Sin Inf* 2015; 45(3): 398.
31. Jin LP and Dong J. Ensemble deep learning for biomedical time series classification. *Comput Intell Neurosci* 2016; 2016: 6212684.
32. Pedregosa F, Varoquaux G, Gramfort A, et al. Scikit-learn: machine learning in Python. *J Mach Learn Res* 2011; 12: 2825–2830.
33. Buitinck L, Louppe G, Blondel M, et al. API design for machine learning software: experiences from the scikit-learn project. In: *ECML PKDD workshop: languages for data mining and machine learning*, Prague, 23 September 2013, pp.108–122.
34. LeCun Y, Bottou L, Bengio Y, et al. Gradient-based learning applied to document recognition. *P IEEE* 1998; 86(11): 2278–2324.
35. Bouvrie J. Notes on convolutional neural networks, 2006, http://cogprints.org/5869/1/cnn_tutorial.pdf

36. Nair V and Hinton GE. Rectified linear units improve restricted Boltzmann machines. In: *Proceedings of the 27th international conference on machine learning (ICML-10)*, Haifa, Israel, 21–24 June 2010, pp.807–814. Madison, WI: Omnipress.
37. Maas AL, Hannun AY and Ng AY. Rectifier nonlinearities improve neural network acoustic models. In: *Proc. ICML*, vol. 30, 2013, http://ai.stanford.edu/~amaas/papers/relu_hybrid_icml2013_final.pdf
38. Hochreiter S, Bengio Y, Frasconi P, et al. Gradient flow in recurrent nets: the difficulty of learning long-term dependencies, 2001, http://www.bioinf.jku.at/publications/older/ch7.pdf
39. LeCun YA, Bottou L, Orr GB, et al. Efficient BackProp. In: Montavon G, Orr G and Müller KR (eds) *Neural networks: tricks of the trade*. Berlin: Springer, 2012: pp.9–48.
40. He K, Zhang X, Ren S, et al. Identity mappings in deep residual networks. In: *European conference on computer vision*, Amsterdam, 8–16 October 2016, pp.630–645. Cham: Springer.
41. Kingma D and Ba J. Adam: a method for stochastic optimization. arXiv preprint arXiv:14126980, 2014.
42. Avriel M. *Nonlinear programming: analysis and methods*. Mineola, NY: Dover Publications, 2003.
43. Theano Development Team. Theano: a Python framework for fast computation of mathematical expressions. arXiv preprint arXiv:1605.02688, 2016, http://arxiv.org/abs/1605.02688
44. Dieleman S, Schlüter J, Raffel C, et al. Lasagne: first release, 2015, DOI:10.5281/zenodo.27878, http://dx.doi.org/10.5281/zenodo.27878
45. Powers DM. Evaluation: from precision, recall and f-measure to roc, informedness, markedness and correlation, 2011, https://bioinfopublication.org/files/articles/2_1_1_JMLT.pdf