Dynamic Convolutional Neural Network for Activity Recognition

Chih-Hsiang You* and , Chen-Kuo Chiang†
* National Chung Cheng University, Taiwan
E-mail: j323919@gmail.com
† National Chung Cheng University, Taiwan
E-mail:ckchiang@cs.ccu.edu.tw

***Abstract*—In this paper, a novel Dynamic Convolutional Neural Network (D-CNN) is proposed using sensor data for activity recognition. Sensor data collected for activity recognition is usually not well-aligned. It may also contains noises and variations from different persons. To overcome these challenges, Gaussian Mixture Models (GMM) is exploited to capture the distribution of each activity. Then, sensor data and the GMMs are screened into different segments. These segments form multiple paths in the Convolutional Neural Network. During testing, Gaussian Mixture Regression (GMR) is applied to dynamically fit segments of test signals into corresponding paths in the CNN. Experimental results demonstrate the superior performance of D-CNN to other learning methods.**

I. Introduction

With the recent advance of wearable devices, human activity recognition can be achieved by collecting sensor data via wearable devices. Activity Recognition plays an important roles in the daily life and has significant impact to many applications, such as daily lifelog [1], health care [2], elderly care [3], and personal fitness [4].

Human activity recognition has been an important problem in the research field of computer vision. The sensor data from triaxial accelerometers is exploited greatly to recognize human activities. To process sensor data, feature extraction is usually adopted as the first step. Traditionally, it captures statistical information through the mean, variance or entropy to extract features. Other statistical methods in the frequency domain, such as FFT [5], are also widely used. However, these methods are applicable to single action identification and ineffective to the recognition of multiple actions [6]. Principal Component Analysis (PCA) is a common technique to capture features of sensor data. Since it can only capture the linear structure of feature space, we need other methods for more complex activities in the nonlinear feature space.

Deep learning is regarded as one of the machine learning methods with the greatest potentials to solve many computer vision problems. Convolution Neural Network (CNN) is one of the most popular method in deep learning [7], [8]. Features extracted by CNN from sensor signals for activity recognition has two advantages. It can capture local dependency of signals and can also process signals of different scales, such as frequency or amplitude.

In this paper, a novel Dynamic Convolutional Neural Network (D-CNN) is proposed using sensor data for activity recognition. One of the challenges using sensor data for activity recognition is the data alignment. During data collection, the starting and ending time, as well as the speed, to perform activities may be different. It also contains noises and variations when the activity is performed by different persons. To overcome these challenges, Gaussian Mixture Models (GMM) is exploited to capture the distribution of each activity in the training process. Then, data partition and channel fitting steps are applied to fit each segment of input signal to the corresponding channels in the CNN via Gaussian Mixture Regression (GMR). Our method is called *Dynamic Convolutional Neural Network* because the entire input signal is partitioned into segments. These segments are dynamically assigned to different channels in CNN based on the nearest segments of GMM models.

The main contribution of our method is to dynamically assign signal segments to corresponding channels in CNN. This approach helps fit similar input signals to the correct channels of CNN under the conditions of noises, different speed of activity and signals that are not well-aligned. To partition signals into segments and dynamically assign them to the corresponding channels, our method has the potential to recognize signals containing multiple activities. For example, it can deal with the signals containing walking, jumping and climbing stairs within the same signals.

II. Proposed Method

A. Preprocessing and Feature Extraction

The input signals of our method are retrieved from three-axis accelerometers. To preprocess the raw data, a fixed-size sliding window is applied to the raw data. Then, a median filter (size: 3) is used to remove the signal noises. Last, we normalize the signals to the range [0, 1]. To extract features from input signals,low-pass filter [9][10] is applied to extract the *gravity* and *body acceleration* as two features of the three-axis signals. The gravity feature is the signal associated with the gravitational acceleration. The signals without gravitational acceleration corresponds to the body feature. The system structure of our method is depicted in Fig. 1.

B. GMM-GMR model

GMM-GMR [11] is exploited to model the distributions of different activities. This overcomes the within-class variations

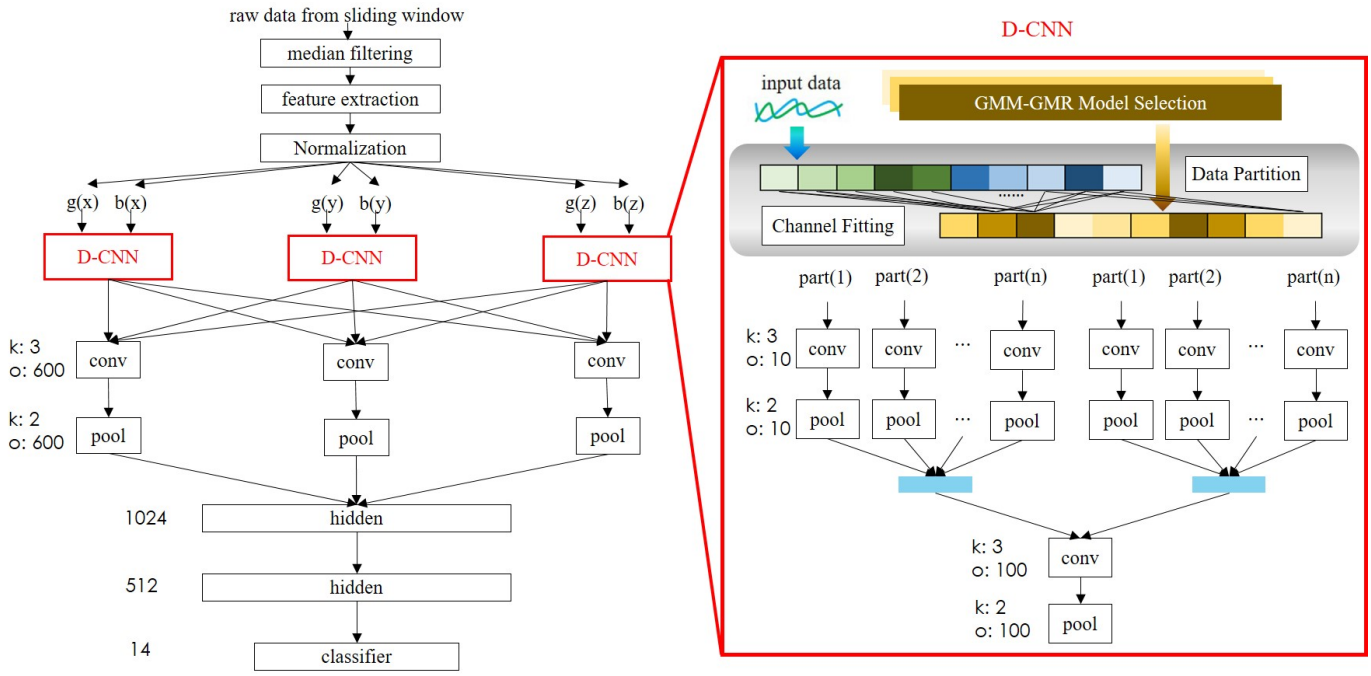

Fig. 1: System structure of our method. k is kernel size (1 by k). o is number of out feature map. $g(x)$ is gravity feature of x-axis signal. $b(x)$ is body feature of x-axis signals. $g(y)$, $b(y)$, $g(z)$, b(z) are those of y- and z-axis.

when collecting sensor data for one activity.

Let $g_t = (t, g_x(t), g_y(t), g_z(t))$, $g_t \in \mathbf{R}^4$, $t = 1, ..., T$, where time size is T. $b_t = (t, b_x(t), b_y(t), b_z(t))$, $b_t \in \mathbf{R}^4$, $t = 1, ..., T$. GMM with K components are used to model g_t and b_t. Let $\mu_k \in \mathbf{R}^4$ and $\Sigma_k \in \mathbf{R}^{4x4}$ the mean vector and the covariance matrix for g_t, where $k = 1, ..., K$. We use GMR to determine the the mean and the covariance matrix at time t for k-th GMM component. We first separate the temporal and acceleration values in μ_k and Σ_k as follows:

$$\mu_k = \left\{\mu_k^t, \mu_k^a\right\}. \qquad \Sigma_k = \begin{pmatrix} \Sigma_k^{tt} & \Sigma_k^{ta} \\ \Sigma_k^{at} & \Sigma_k^{aa} \end{pmatrix}. \tag{1}$$

The expected mean acceleration $\hat{\mu}_k^a$ of the k component at time index t and the associated covariance matrix $\hat{\Sigma}_k^{aa}$ can be defined as:

$$\begin{cases} \hat{\mu}_k^a = \mu_k^a + \Sigma_k^{at}(\Sigma_k^{tt})^{-1}(t - \mu_k^t) \\ \hat{\Sigma}_k^{aa} = \Sigma_k^{aa} - \Sigma_k^{at}(\Sigma_k^{tt})^{-1}\Sigma_k^{ta} \end{cases} \tag{2}$$

Then, the $\hat{\mu}_k^a$ and $\hat{\Sigma}_k^{aa}$ are mixed by the probability β_k of the k component at time index t to compute the expected acceleration μ^a and covariance matrix Σ^{aa} at time index t, as follows:

$$\beta_k = \frac{p(k)p(t|k)}{\sum_{j=1}^{K} p(j)p(t|j)} = \frac{\pi_k \mathcal{N}(t; \mu_k^t, \Sigma_k^{tt})}{\sum_{j=1}^{K} \pi_j \mathcal{N}(t; \mu_j^t, \Sigma_j^{tt})} \tag{3}$$

$$\mu^a = \sum_{k=1}^{K} \beta_k \hat{\mu}_k^a. \qquad \Sigma^{aa} = \sum_{k=1}^{K} \beta_k^2 \hat{\Sigma}_k^{aa}. \tag{4}$$

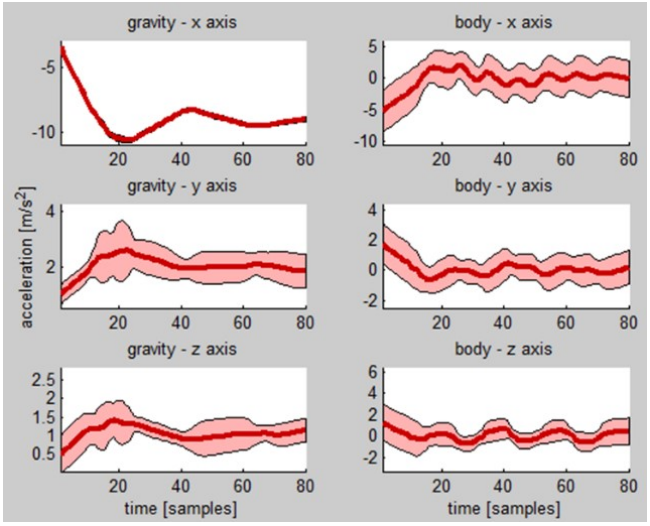

Fig. 2: GMM-GMR model of climb stairs activity in WHARF dataset. Red line is acceleration value of every time point. Pink area is standard deviation of every time point.

where $\mathcal{N}$ is gaussian distribution function. Therefore, we can calculate the mean and the associated covariance matrix of acceleration at every the time t in the sequence T to build our GMM-GMR model as Fig. 2.

C. Dynamic Assignment

Once the GMM-GMR model is trained, the dynamic assignment approach is adopted to fit the input signal to the GMM-GMR model in a segment basis. The dynamic assignment

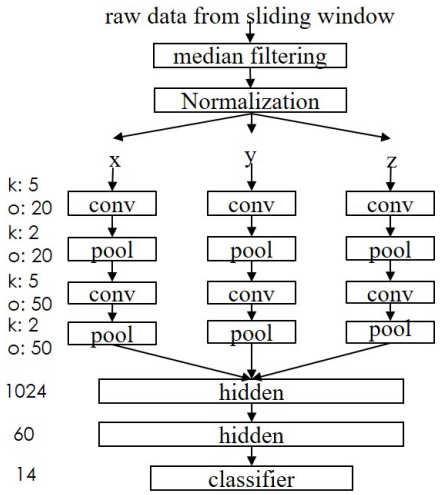

Fig. 3: CNN model

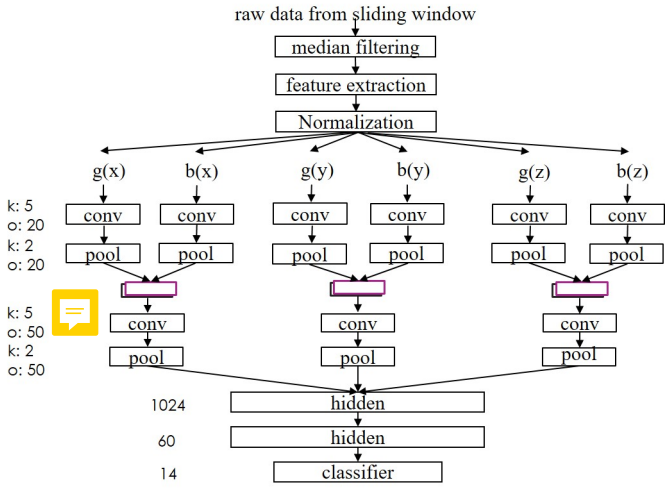

Fig. 4: GB-CNN model

contains two steps: *Data Partition* and *Channel Fitting*.

1) Data Partition: The GMM-GMR model is trained for each activity class. Then, the model is partitioned into N parts which correspond to the N channels in the D-CNN. The features are also partitioned into N parts. By channel fitting, features which are similar to the same model part go to the same channel in the D-CNN.

2) Channel Fitting: The distance of partitioned features and the model part can be calculated by the Mahalanobis distance. Denote x_t the triaxial acceleration signal at the time t, the Mahalanobis distance between the signal at the time t and the model part can be defined as:

$$d_t = \sqrt{(x_t - \mu_t^a)^T (\Sigma_t^{aa})^{-1} (x_t - \mu_t^a)} \tag{5}$$

where μ_t^a and Σ_t^{aa} are the mean and the covariance matrix of the model at time t. So the distance between the partitioned features and the model part can be computed as:

$$d = \frac{1}{n} \sum_{t=1}^{n} d_t \tag{6}$$

where n is the size of the partitioned feature.

The N model parts correspond to N channels of CNN. Here, *channel* is referred to as one path containing convolution and pooling operations in the CNN. Then, the results of N channels are concatenated to build the feature map. Last, two feature maps of gravity and body are used combined as input to another convolution and pooling operations to capture the correlation between two features. The process of D-CNN is depicted by the red frame in the right of Fig. 1.

D. Variations of CNN Models

There are several variations of CNN models by considering the correlation between body and gravity features or the correlations among tri-axis signals. In this sub-section, we will introduce different models and compare these variations in the experiments.

1) Basic CNN Model: The basic $CNNmodel$ uses raw data without extracting features. In addition, the signals of x-, y- and z-axis has their own path for convolution and pooling operations. The output of these paths are then concatenated as input to the hidden layer. The basic CNN model is depicted in Fig. 3.

2) GB-CNN Model: By extracting gravity (G) and body (B) features, in the GB-CNN model, gravity and body features from the same axis are combined for common convolution and pooling operations. Then, the outputs from three axes are concatenated and fed into the hidden layers. The GB-CNN model is depicted in Fig. 4.

3) 3GB-CNN Model: In order to capture the correlations among signals of three axes, a fully connected relations are built for signals of three axes and three paths of convolution and pooling. Compared to GB-CNN, the red blocks in Fig. 5 with the fully connected relations depict the difference from GB-CNN.

4) D-CNN Model: To further improve the 3GB-CNN model, the conventional convolution and pooling operations are replaced with the N channels and the GMM-GMR models. Therefore, similar signals can be assigned to the same channel of D-CNN to reduce the impact of data noises and ill alignment. The D-CNN model is depicted in Fig. 1.

III. EXPERIMENTS

A. Dataset and Settings

We use Wearable Hutman Activity Recognition Folder (WHARF) dataset [11] for our experiments. It contains 14 kinds of activity signals collected by triaxial accelerometer from daily life. The sample rate of the sensor is 32 Hz. We use sliding window (size: 80, overlap: 50 %) for preprocessing. In each class, we take 120 data for training and the others for testing. The models of NN, CNN, GB-CNN, 3GB-CNN and D-CNN use the same parameter settings with learning rate 0.01, momentum 0.9, and weight decay 0.0005.

TABLE I: The classification accuracy on the WHARF dataset using different methods

Activities	Method						
	CNN	GB-CNN	3GB-CNN	SVM	KNN	NN	**D-CNN**
Brush teech	85.03	85.40	87.31	87.34	71.55	68.75	**90.87**
Climb stairs	50.13	56.39	**59.91**	0.80	41.69	32.98	57.26
Comb hair	57.92	67.92	71.28	64.07	30.73	42.55	**83.01**
Descend stairs	74.52	64.96	77.04	46.15	65.87	71.15	**80.72**
Drink glass	57.45	66.31	66.88	28.97	62.38	62.26	**76.40**
Eat meat	76.42	77.67	84.97	83.77	88.21	89.74	**91.50**
Eat soup	83.33	**92.11**	89.24	85.71	83.33	80.95	82.32
Getup bed	23.83	33.15	39.20	11.52	32.16	23.72	48.04
Liedown bed	21.60	43.59	**50.00**	14.40	18.40	32.00	49.31
Pour water	54.68	70.85	72.34	39.61	51.43	56.23	**77.85**
Sitdown chair	42.94	**65.83**	68.03	27.15	56.79	50.97	67.80
Standup chair	39.57	60.41	65.62	28.88	48.40	58.82	**70.45**
Use telephone	67.63	79.09	79.12	69.29	72.61	71.37	**82.91**
Walk	61.62	72.49	74.90	**80.84**	60.05	68.80	75.92
Overall accuracy	56.41	67.01	70.33	50.27	55.78	57.71	**74.58**

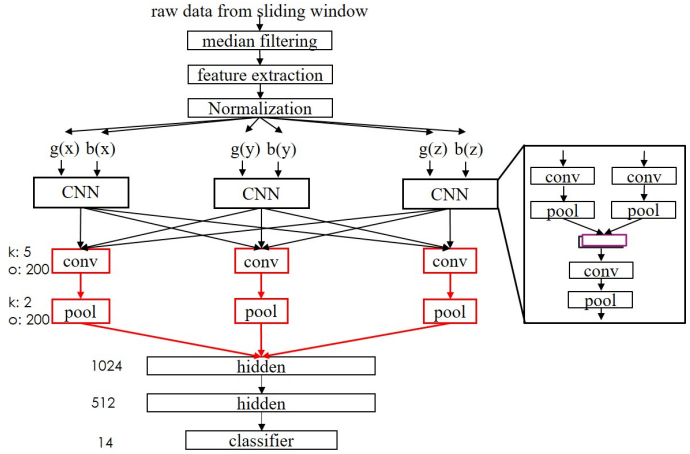

Fig. 5: 3GB-CNN model

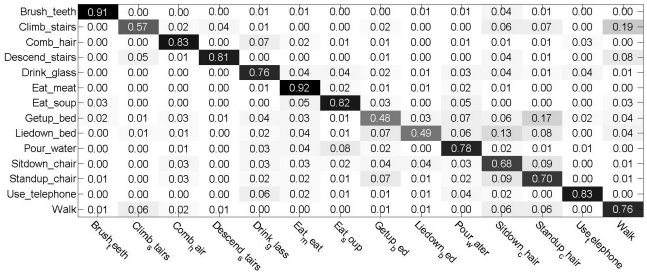

Fig. 6: Confusion matrix of D-CNN on the WHARF dataset.

B. Classification Accuracy

In our experiments, we first compare the learning models CNN, GB-CNN, 3GB-CNN to the proposed D-CNN. The classification accuracy is presented in Table I. It shows that the overall accuracy can be improved up to 10.6% by exploiting gravity and body features in GB-CNN compared to CNN. We can also note that by capturing the correlation of tri-axis signals 3GB-CNN, the overall accuracy can be further improved by 3.32 %. By partitioning the signals into 4 parts in the D-CNN, the overall accuracy is 4.25 % higher than that of 3GB-CNN. The confusion matrix of D-CNN is presented in Fig. 6. Overall, the proposed D-CNN enhances the overall accuracy up to 18.17% compared to the basic CNN framework.

We also compare to other learning methods, including SVM, K-Nearest Neighbor (KNN) and Neural Network (NN). The three-axis signals are concatenated into a single vector as input to train the SVM and used for KNN and NN. The NN has two hidden layers, which contain 1024 and 512 nodes, respectively. According to the results in Table I, our method has the highest overall accuracy and has the highest accuracy of most categories.

IV. Conclusions

A dynamic CNN (D-CNN) is proposed in this paper. D-CNN can assign similar signal parts to the same CNN channel. Therefore, it can better deal with the problem of data noise, alignment and other data variations. Demonstrated by the experiments, the results of classification accuracy have shown that D-CNN is very effective for activity recognition using sensor data. In the future, we will extend our framework to complex activity and signals containing multiple activities.

References

[1] Snehal Chennuru, Peng-Wen Chen, Jiang Zhu, and JoyYing Zhang, "Mobile lifelogger recording, indexing, and understanding a mobile user life," in *Mobile Computing, Applications, and Services*, pp. 263–281. 2012.

[2] Pang Wu, Jiang Zhu, and Joy Ying Zhang, "Mobisens: A versatile mobile sensing platform for real-world applications.," *MONET*, , no. 1, pp. 60–80, 2013.

[3] Pang Wu, Huan-Kai Peng, Jiang Zhu, and Ying Zhang, "Senscare: Semi-automatic activity summarization system for elderly care," in *Mobile Computing, Applications, and Services*, vol. 95, pp. 1–19. 2012.

[4] K. Forster, D. Roggen, and G. Troster, "Unsupervised classifier self-calibration through repeated context occurences: Is there robustness against sensor displacement to gain?," in *Wearable Computers, 2009. ISWC '09. International Symposium on*, Sept 2009, pp. 77–84.

[5] A. Krause, D.P. Siewiorek, A. Smailagic, and J. Farringdon, "Unsupervised, dynamic identification of physiological and activity context in wearable computing," in *Wearable Computers, 2003. Proceedings. Seventh IEEE International Symposium on*, Oct 2003, pp. 88–97.

[6] Tâm Huynh and Bernt Schiele, "Analyzing features for activity recognition," in *Proceedings of the 2005 Joint Conference on Smart Objects and Ambient Intelligence: Innovative Context-aware Services: Usages and Technologies*. 2005, sOc-EUSAI '05, pp. 159–163, ACM.

[7] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems 25*, pp. 1106–1114. 2012.

[8] Pierre Sermanet, Koray Kavukcuoglu, Soumith Chintala, and Yann LeCun, "Pedestrian detection with unsupervised multi-stage feature learning," *CoRR*, vol. abs/1212.0142, 2012.

[9] D.M. Karantonis, M.R. Narayanan, M. Mathie, N.H. Lovell, and B.G. Celler, "Implementation of a real-time human movement classifier using a triaxial accelerometer for ambulatory monitoring," *Information Technology in Biomedicine, IEEE Transactions on*, vol. 10, no. 1, pp. 156–167, Jan 2006.

[10] G. Krassnig, D. Tantinger, C. Hofmann, T. Wittenberg, and M. Struck, "User-friendly system for recognition of activities with an accelerometer," in *Pervasive Computing Technologies for Healthcare (PervasiveHealth), 2010 4th International Conference on-NO PERMISSIONS*, March 2010, pp. 1–8.

[11] B. Bruno, F. Mastrogiovanni, A. Sgorbissa, T. Vernazza, and R. Zaccaria, "Analysis of human behavior recognition algorithms based on acceleration data," in *Robotics and Automation (ICRA), 2013 IEEE International Conference on*, May 2013, pp. 1602–1607.