2018 16th International Conference on Frontiers in Handwriting Recognition

# Introducing Local Distance-Based Features to Temporal Convolutional Neural Networks

Brian Kenji Iwana*, Minoru Mori†, Akisato Kimura‡, Seiichi Uchida*
*Department of Advanced Information Technology, Kyushu University, Fukuoka, Japan
†Department of Information and Computer Sciences, Kanagawa Institute of Technology, Atsugi, Japan
‡Nippon Telegraph and Telephone, Tokyo, Japan
brian@human.ait.kyushu-u.ac.jp, mmori@ic.kanagawa-it.ac.jp, kimura.akisato@lab.ntt.co.jp, uchida@ait.kyushu-u.ac.jp

***Abstract*—In this paper, we propose the use of local distance-based features determined by Dynamic Time Warping (DTW) for temporal Convolutional Neural Networks (CNN). Traditionally, DTW is used as a robust distance metric for time series patterns. However, this traditional use of DTW only utilizes the scalar distance metric and discards the local distances between the dynamically matched sequence elements. This paper proposes recovering these local distances, or DTW features, and utilizing them for the input of a CNN. We demonstrate that these features can provide additional information for the classification of isolated handwritten digits and characters. Furthermore, we demonstrate that the DTW features can be combined with the spatial coordinate features in multi-modal fusion networks to achieve state-of-the-art accuracy on the Unipen online handwritten character datasets.**

## I. Introduction

Dynamic Time Warping (DTW) [1] is a dynamic programming solution to finding the optimal match between elements for the use of a distance metric between time series patterns. DTW has been shown to be useful as a distance metric due to it being relatively fast and robust to temporal distortions and rate differences [2]. DTW has been used for many time series applications and been incorporated in many time series recognition methods. For example, a common usage of DTW is as a distance metric for $k$-Nearest Neighbor ($k$-NN) [3], [4]. Other applications of DTW include using it with Support Vector Machines (SVM) [5], Dissimilarity Space Embedding (DSE) [6], and Hidden Markov Models (HMM) [7].

To be used as a distance metric, DTW is calculated by summing the local distance between the optimally matched sequence elements. The elements are matched by estimating the minimal path on a cost matrix given constraints. However, when using DTW as a distance metric, only the resulting summation of local distances is used. While this summation value is useful for the global distance metric between patterns, the local distances and the matching information is discarded. It is possible that there is useful information within the connections between the matched elements that is not reflected in the summation value.

The discarded information from a DTW can convey the relationship between the elements of the samples and matched sequence elements of the prototypes. By capturing the local distances between matched elements, a sequence of local distance-based features, or *DTW features*, is created. The DTW features can then be used as a time series in time series recognition methods.

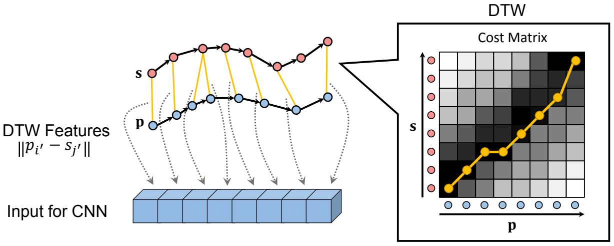


Fig. 1. Extraction of DTW features from a DTW calculation. Given a prototype sequence $\mathbf{p}$ and a sample sequence $\mathbf{s}$, the cost matrix is the pairwise local distances between each element in the time series with dark boxes indicating low cost and light boxes indicating high cost. The yellow warping path on the cost matrix is the estimated minimal path.

Recently, artificial neural networks (ANN) and neural models have achieved state-of-the-art results in pattern recognition and machine learning [8]. Convolutional Neural Networks (CNN) [9] in particular have had many successes in the image domain [10], [11]. Recurrent Neural Networks (RNN) [12] and more specifically Long Short-Term Memory RNNs (LSTM) [13] have had success in the time series and sequence domains [14], [15]. While RNNs traditionally have been used for time series patterns, there have been many works which use CNNs for temporal data and it has been shown that temporal CNNs can be competitive and sometimes even more effective than RNNs [16]. In addition, the results presented in this paper reflect these findings.

In this paper, we propose the idea of using the sequence of DTW features as a time series for the input of a temporal CNN, as shown in Fig. 1. The contribution of this paper is threefold. First, we show that the novel DTW features recovered from the DTW calculation can be used for classification by CNN. Second, we analyze different aspects of using CNNs with time series data, such as 1D versus 2D convolutions and convolutional window size. Third, we combine the DTW features and the original spatial coordinates, or *spatial features*, in fusion networks to acquire state-of-the-art results.

978-1-5386-5875-8/18/$31.00 ©2018 IEEE
DOI 10.1109/ICFHR-2018.2018.00025

## II. Related Work

Temporal CNNs are a natural extension of CNNs to time series using a similar function of Time Delay Neural Networks (TDNN) [17]. The most common application of temporal CNNs is to use convolutions that stride in one direction, or 1D convolutions. In this way, the convolution can encompass multivariate data but only moves across the time steps. These 1D temporal CNNs have been used many applications and time series data types [18]–[21] and have shown to be competitive and at times, more effective than RNNs, LSTMs, and Gated Recurrent Units (GRU) [16]. In a recent advancement, WaveNet [22], has produced impressive results for audio generation by implementing dilated causal convolutional layers. In addition, CNNs have been used for other non-image structural domains such as graphs [23].

## III. Local Distance-based Features

The primary goal of the proposed method is to reuse the discarded information from DTW and use it for classification by CNN. In this section, we detail the method of extracting the DTW features.

### A. Dynamic Time Warping

Discrete time series recognition is not a straightforward task due to challenges in overcoming temporal distortions such as rate differences, translations in time, and variable lengths. DTW is a popular method used to consider these challenges and provide an effective distance metric between time series [2]. Specifically, DTW aligns similar features of the time series by warping the sequences in the time dimension. In this way, elements of the time series are elastically matched to minimize the summation of local distances between those matched elements, while under constraints to maintain the temporal qualities of the time series.

Namely, DTW can be found between two time series, sequence $\mathbf{p} = p_1, \dots, p_i, \dots, p_I$, referred to as the *prototype*, and sequence $\mathbf{s} = s_1, \dots, s_j, \dots, s_J$, referred to as the *sample*, where $i$ and $j$ are the indices of sequence elements $p_i$ and $s_j$ with length $I$ and $J$, respectively. For online handwriting, the sequence elements are pen tip captured spatial coordinates, which we refer to as spatial features.

Given $\mathbf{p}$ and $\mathbf{s}$, DTW is defined as the global summation of the local distances between pairwise element matches, or:

$$\mathrm{DTW}(\mathbf{p}, \mathbf{s}) = \sum_{(i',j') \in \mathcal{M}} ||p_{i'} - s_{j'}||, \tag{1}$$

where $(i', j')$ is a pairwise match determined by dynamic programming over a local cost matrix and $\mathcal{M}$ is the set of all matches. The indices $i'$ and $j'$ correspond to the original indices $i$ and $j$ of $\mathbf{p}$ and $\mathbf{s}$, respectively. It should be noted that the set $\mathcal{M}$ may contain duplicate or skipped indices.

In this paper, we utilize the asymmetric slope constraint defined by the recurrent function:

$$D(i, j) = ||p_i - s_j|| + \min_{j' \in \{j, j-1, j-2\}} D(i-1, j'), \tag{2}$$

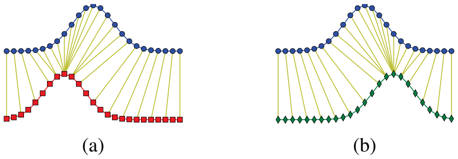


Fig. 2. Illustration of two DTW calculations with a similar result but different matching relationships. The prototype pattern is represented in blue and the sample patterns are represented in (a) red and (b) green. Yellow lines are the pairwise matches determined by DTW.

where $D(i, j)$ is the cumulative sum of the $i$-th and $j$-th elements. The global DTW distance is determined by the cumulative sum at $D(I, J)$. This asymmetric slope constraint is used due to the special property that the number of pairwise matches in $\mathcal{M}$ is exactly equal to the number of elements $I$ in the prototype $\mathbf{p}$. Additionally, $i'$ always corresponds to the same time step as $i$ in the original prototype sequence. However, $j'$ can correspond non-linearly to the time steps $j$ of the sample. This means that no matter how many elements $\mathbf{s}$ contains, the number of matches in $\mathcal{M}$ will always equal the number of elements in $\mathbf{p}$.

### B. DTW Features

Traditionally, only the global distance $D(I, J)$ is used for pattern recognition methods and the individual local distances in Eq. (1) of the warping path are discarded. However, when strictly using the global distance, information about the structure of the comparison patterns can be lost. For example, Fig. 2 illustrates a case where the DTW distance is similar, but the patterns from Figs. 2a and 2b are clearly different. Thus, preserving the information about the specific matches could be useful to discriminate patterns.

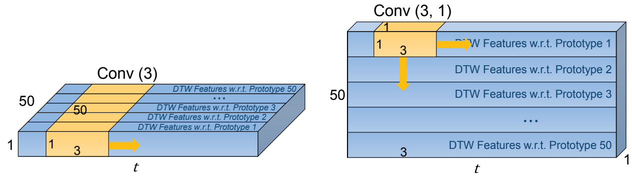
By preserving the information about the specific matches in the DTW calculation, new DTW features can be extracted. Given the slope constraint in Eq. (2) with prototypes of a fixed length, the number of matches in $\mathcal{M}$ would be fixed. In this way, a sequence of fixed length can be extracted from the DTW calculation by considering the local distances between matches as a time series. Namely, the time series of DTW features is defined as:

$$\mathbf{m}_n = ||p_{n,1} - s_1||, \dots, ||p_{n,i'} - s_{j'}||, \dots, ||p_{n,I'} - s_{J'}||, \tag{3}$$

where $(i', j') \in \mathcal{M}$ and $\mathbf{m}_n$ is the time series of DTW features relating to the $n$-th prototype $\mathbf{p}_n$ with a prototype set of $N$ number of prototypes. The result is a $N$-dimensional time series of a fixed length $I$ based on the local distances of matched elements between a sample and each prototype.

## IV. Temporal Convolutional Neural Networks

CNNs are feedforward neural networks usually contain three components, convolutional layers, pooling layers, and fully-connected layers. Typically, they are used with structural data, most notably images, due to the ability learn structural qualities while maintaining some shape and location invariance. The core component of a CNN is the convolutional


Conv (3)
50
50
1
1
3
t
DTW Features w.r.t. Prototype 50
DTW Features w.r.t. Prototype 3
DTW Features w.r.t. Prototype 2
DTW Features w.r.t. Prototype 1
Conv (3, 1)
1
1
3
50
DTW Features w.r.t. Prototype 1
DTW Features w.r.t. Prototype 2
DTW Features w.r.t. Prototype 3
...
DTW Features w.r.t. Prototype 50
3
1
t

(a) 1D Convolution

(b) 2D Convolution

Fig. 3. Comparison of a (a) 1D convolution and (b) 2D convolution of the input layer using a sequence created from DTW features. In (a), the 1D convolution considers each DTW w.r.t. a prototype as a dimensions of the input and strides in the time dimension $t$. In (b), the 2D convolution shares weights and strides both in the time dimension and across prototypes.

layer, which applies shared weights, or a *convolutional filter*, across an input. The application of the convolution on the input is similar to a feature extractor [9] and the result is a feature map. Furthermore, pooling layers are used to downsample the feature maps in order to reduce redundancy and introduce small translation and size invariance. Finally, the fully-connected layers and the output layers are used to learn a non-linear function from the high-level features of the high-level convolutional layer.

The general idea of the proposed method is to construct input sequences of DTW features and use them with a CNN. However, using a CNN for time series can be done in two ways, a traditional 1D convolution CNN and a proposed 2D convolution CNN.

### A. CNNs with a 1D Convolution

The traditional method of using a CNN with time series patterns is to use a 1D convolution in the convolutional layers. Shown in Fig. 3a, the convolution strides across the time series in the time step dimension and considers the dimensionality of the time series as depth (similar to the interaction of channels for images). This is the equivalent of using a convolution over a signal. In this method, parameters are shared among time steps and the 1D convolution CNN is able to overcome small temporal distortions by learning high-level features through pooling.

### B. CNNs with a 2D Convolution

While a 1D convolution CNN shares parameters across time steps, it is possible to share parameters across both time steps and prototypes. To accomplish this, we propose considering the time series as a matrix and employ traditional 2D convolutions for the convolutional layer. Figure 3b is a demonstration of this using the proposed DTW features. Using a 2D convolution in this manner is possible if the window size of the 2D convolution has a height of 1. A convolutional window size with a height of 1 is used because order of the DTW features with respect to each prototype is arbitrary. However, the 2D convolution is still useful because unlike a 1D convolution with different parameters for each DTW feature channel, parameters are shared with respect to the different prototypes.

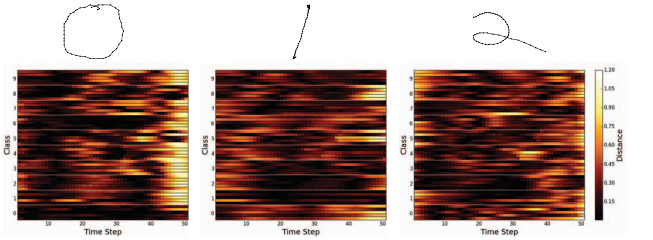


Fig. 4. Examples of DTW feature sequences using 50 prototypes. (above) is the sample digit and (below) is the resulting DTW feature sequence. The prototypes are sorted by class for visualization purposes.

## V. Comparative Results between Spatial Features and DTW Features

### A. Dataset

Isolated online handwritten characters serve as a valuable source of time series data due to having distinct classes yet still having a high variation within those classes. For the experiments, we used the Unipen multi-writer 1a (digits), Unipen 1b (uppercase alphabet), and Unipen 1c (lowercase alphabet) datasets [24]. The patterns in the datasets consist of about 15,000 time series made of 2D pen tip coordinates and was collected by the International Unipen Foundation (iUF).[1] This is a well established source of data for time series recognition.

For the experiments, the dataset was divided into three sets, a training set, a test set, and a prototype set. For the Unipen 1a evaluation, the test set contained 1,300 patterns and the training set contained 11,650 patterns with an even class distribution. The prototype set contained 50 patterns with 5 patterns from each class selected at random. For the Unipen 1b and 1c evaluations, 1,235 patterns were used for the test set, 11,063 patterns were used for the training set, and 2 of each class were used for the prototype set.

Furthermore, the all patterns were preprocessed by scaling and resampling. The patterns were scaled so that each pattern could fit within the bounds of a square defined by $(-1,-1)$ and $(1,1)$. Finally, patterns were resampled to 50 elements in length to fix the DTW feature sequences to 50 elements and to be able to use the patterns for the spatial feature trials.

To prepare the experiments with DTW features, DTW was calculated between the patterns in the prototype set and each pattern in the training and test sets. The DTW features were extracted as described in Section III-B and $N$-dimensional time series with length $I = 50$ were created. Figure 4 is a visualization of example sample patterns encoded into DTW feature sequences. Lastly, the DTW features were normalized to be -1 to 1.

### B. Settings and Evaluation

*1) CNN Evaluations:* In order to compare the proposed method of using DTW features with CNNs to the pen tip features, we trained four different CNNs architectures using

[1] URL: http://www.unipen.org/home.html

input sequences of one of the two feature types. *1D CNN (3)* and *1D CNN (5)* are 1D convolutional CNNs using a convolutional window size of 3 and 5, respectively. Different sized windows were used to reveal the effect of window size on the results. Similarly, *2D CNN (3, 1)* and *2D CNN (5, 1)* are 2D convolution CNNs with window sizes (3, 1) and (5, 1), respectively. The *Spatial Features* CNNs were trained using only the 2D spatial features as a sequence and the *DTW Features* CNNs used the DTW feature sequences.

The CNNs used for the experiment are constructed of three convolutional layers with accompanying max pooling layers, two fully-connected layers, and a softmax output. The convolutional layers have 64, 128, and 256 nodes, respectively, the fully-connected layers were set to 1024 nodes, and softmax had an output respective to the number of classes in the task. Each of the hidden layers use Rectified Linear Unit (ReLU) activations and dropout with a keep probability of 0.5 was used proceeding the fully-connected layers. The networks were trained using Adaptive Moment Estimation (Adam) [25] with an initial learning rate of 0.0001 and with batches of 100 for 100,000 iterations.

*2) LSTM Comparisons:* The LSTMs were constructed to be fair comparisons with the same hyperparameters and structures as the CNNs. Namely, the LSTMs consisted of three stacked LSTM layers with an initial forget bias of 1.0, two fully-connected layers with 1024 nodes, and a softmax output layer. The fully-connected layers use ReLU activations and use dropout with a keep probability of 0.5. Similar to the CNN evaluations, the LSTM evaluations were optimized using Adam optimizer with batches of 100 for 100,000 iterations.

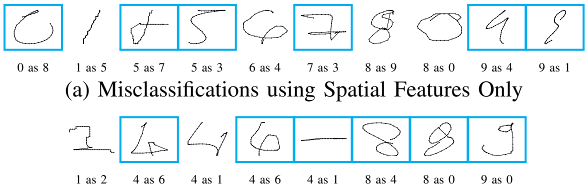
*3) Reported Comparisons:* In addition to the LSTM evaluation, we report the state-of-the-art results found in literature. *HMM-CSDTW* [7], *DAG-SVM-GDTW* [5], and *OnSNT* [26] are established classical baseline methods based on an HMM, an SVM, and an $n$-tuple classifier, respectively. *DTW-NN* [27] and *Google* [28] are the latest state-of-the-art solutions and are based on deep neural networks.

### C. Results

Table I reports the result of the experiments as well as the results from literature. The results show that using the DTW features with 2D CNN (5, 1) achieved the highest accuracy for Unipen 1a and 1c with accuracies of 98.77% and 97.73%, respectively. Conversely, the Unipen 1b experiment had a higher accuracy for the spatial features. However, all of the CNNs, both DTW features and spatial features, had a higher accuracy than the LSTM evaluations and all of the reported results with exception of Google for Unipen 1a. It is also interesting to note that the 2D CNN trials generally performed better than the 1D CNNs despite 1D CNNs being a standard practice for temporal CNN models. From these results, we can see that the DTW features successfully encode information about the underlying sample patterns and sometimes even more so than the spatial features.

TABLE I
COMPARISON OF SPATIAL FEATURES AND DTW FEATURES

|  | **Unipen 1a** | **Unipen 1b** | **Unipen 1c** |
|---|---|---|---|
| **Method** | **Acc. (%)** | **Acc. (%)** | **Acc. (%)** |
| **Spatial Features** |  |  |  |
| 1D CNN (3) | 98.00 | 97.65 | 97.33 |
| 1D CNN (5) | 98.38 | 97.81 | 97.41 |
| 2D CNN (3, 1) | 98.08 | 97.14 | 96.76 |
| 2D CNN (5, 1) | 98.08 | **97.89** | 96.76 |
| LSTM | 98.15 | 96.36 | 95.38 |
| **DTW Features** |  |  |  |
| 1D CNN (3) | 98.15 | 97.65 | 96.76 |
| 1D CNN (5) | 98.62 | 97.25 | 96.19 |
| 2D CNN (3, 1) | 98.69 | 97.57 | 97.65 |
| 2D CNN (5, 1) | 98.77 | 97.73 | **97.73** |
| LSTM | 97.62 | 95.71 | 94.33 |
| DAG-SVM-GDTW [5] | 96.2 | 92.4 | 87.9 |
| HMM-CSDTW [7] | 97.1 | 92.8 | 90.7 |
| OnSNT [26] | 98.9 | 95.7 | 92.1 |
| DTW-NN [27] | 96.8 | – | – |
| Google [28] | **99.2** | 96.9 | 94.9 |



(a) Misclassifications using Spatial Features Only

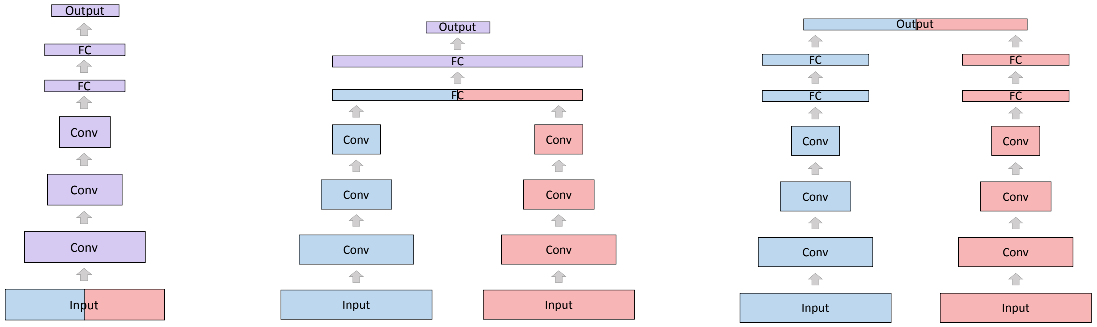
(b) Misclassifications using DTW Features Only

Fig. 5. Unique misclassifications of 1D CNN (5) using (a) spatial features and (b) DTW features. Shared misclassifications are not shown. Digits with blue boxes are misclassifications that were improved by 1D CNN (5) with decision-level fusion.

Figure 5 shows the misclassifications of 1D CNN (5) that were present in the spatial feature experiment but not the DTW feature experiment, and vice versa. The misclassifications in Fig. 5a demonstrate that the 1D CNN (5) is susceptible to noise more so than the misclassifications in Fig. 5b. This is most evident in the “7 as 3” and “8 as 9” in Fig. 5a. The reason for this is due to DTW’s optimal matching which will avoid elements with large distances. Also, interestingly, the DTW features was able to successfully realize the very small upper loop in “8 as 0” in Fig. 5a. Again, it was able to do this because the optimal matching which emphasizes relevant features. However, this also can cause overfitting, such as “1 as 2,” both “4 as 6,” and “8 as 4” in Fig. 5b.

## VI. MULTI-MODAL FUSION

The DTW features and the spatial features are different modalities of data which contain different aspects of the input sample patterns. Thus, it is possible to combine the two features to improve the classification. Combining different types of data in one classifier is referred to as *multi-modal* classification. Using an additional form of data can be used to supplement the discriminative ability of a classifier using only one data modality.



(a) Data-level Fusion (b) Feature-level Fusion (c) Decision-level Fusion

Fig. 6. Multi-modal data fusion in CNNs. *Conv* are convolutional layers and *FC* are fully-connected layers. The blue and red layers indicate the different data types (e.g. spatial features and DTW features) and purple indicates layers post fusion.

However, combining modalities is not a straightforward task. The data can be combined on a input-level, feature-level, or a decision-level, shown in Fig. 6. Input-level fusion concatenates the data modalities and processes them as a single data source. Feature-level fusion combines the different data modalities within the internal representation learned by the model. Finally, decision-level approaches represent the data separately and combine them in a decision rule. The optimal point at which data of different modalities is fused is not well defined.

Fusion neural networks using multi-modal data is an emerging field in deep learning-based machine learning. Neural networks with fusion schemes have been used for combining title and image data [29], and for data from multiple sensors [30], such as multi-source remote sensing [31], activities of daily life [32], and pedestrian detection [33].

### A. Settings and Evaluation

In order to demonstrate that the DTW features can be useful to supplement the raw spatial coordinates of online handwriting, fusion networks were created from the CNNs described in Section III. The previous experiments were repeated using three fusion timings, namely, a data-level fusion CNN, a feature-level fusion CNN, and a decision-level fusion CNN.

*1) Data-level Fusion:* Shown in Fig. 6a, the data-level fusion combined the DTW feature time series and the spatial feature time series into one time series prior to training of the CNN. In this way, a time series with 50 time steps of $(N+2)$-dimensions was fed to the CNN, where $N$ is the number of prototypes and 2 is the extra spatial features. The rest of the hyperparameters are exactly the same as the CNNs described in Section V-B.

*2) Feature-level Fusion:* The architecture of the feature-level fusion network is shown in Fig. 6b. In this fusion network, the DTW feature data and the spatial feature data is split for their own convolutional layers, but are combined right before the first fully-connected layer. Again the hyperparameters and training are the same as before, however, there are twice as many total convolutional nodes due to each modality containing a full set of 64, 128, and 256 nodes.

TABLE II
COMPARISON OF DATA FUSION

| Method | Unipen 1a Acc. (%) | Unipen 1b Acc. (%) | Unipen 1c Acc. (%) |
|---|---|---|---|
| **Data-level Fusion** | | | |
| 1D CNN (3) | 98.00 | 97.25 | 96.28 |
| 1D CNN (5) | 98.15 | 97.17 | 96.44 |
| 2D CNN (3, 1) | 98.69 | 97.41 | 97.09 |
| 2D CNN (5, 1) | 98.23 | 97.33 | 96.84 |
| **Feature-level Fusion** | | | |
| 1D CNN (3) | 98.77 | 97.73 | 96.76 |
| 1D CNN (5) | 98.62 | 97.65 | 96.92 |
| 2D CNN (3, 1) | 98.85 | 97.89 | 97.33 |
| 2D CNN (5, 1) | 98.85 | 97.89 | **97.89** |
| **Decision-level Fusion** | | | |
| 1D CNN (3) | **99.08** | 97.73 | 96.76 |
| 1D CNN (5) | 98.85 | 97.73 | 97.00 |
| 2D CNN (3, 1) | 98.54 | 97.65 | 97.41 |
| 2D CNN (5, 1) | 98.46 | **98.22** | 97.41 |

*3) Decision-level Fusion:* The third fusion timing, decision-level fusion, is shown in Fig. 6c. In this model, two almost complete CNNs, one for each feature type, is used and only combined for the softmax output layer.

### B. Results

The results of the fusion CNN experiments are shown in Table II. The decision-level fusion 1D CNN (3) achieved a near perfect 99.08% accuracy for the Unipen 1a dataset, which is exceptional compared to the results in Table I. In addition, the results from decision-level fusion 2D CNN (5, 1) for Unipen 1b and feature-level fusion 2D CNN (5, 1) for Unipen 1c surpassed all of the other results for their respective datasets. This demonstrates that the proposed method of supplementing time series data with DTW-based distance features to prototypes can achieve a dramatic increase in accuracy.

Table II also reveals that later fusions tended to be more robust. From this, it can be inferred that the modalities interfered with each other when fused early. By fusing at the data-level, the information from the spatial features is dwarfed due to the higher dimensionality of the DTW features. Consequently, the data-level fusion results are similar to the DTW features alone. On the other hand, Fig. 5 provides insight into the reason for the higher decision-level fusion accuracies. In the figure, the decision-level fusion network overcame the weaknesses of the spatial features or the DTW features alone.

## VII. Conclusion

In this paper, we proposed a novel use of the DTW calculation by using it as a method to extract distance-based features. These features, or DTW features, are the local distances between elements matched by DTW. Through extensive experimentation with CNNs, we demonstrated that the DTW features can be used as time series and are useful for classification. Furthermore, we demonstrated that the DTW features can be combined with the original spatial coordinates within multi-modal fusion networks to achieve state-of-the-art results. Using the proposed method, we achieved a 99.08%, 98.22%, and 97.89% accuracy for the Unipen 1a, 1b, and 1c datasets, respectively. These results are dramatic improvements over the results from a CNN on the spatial coordinates alone, the reported results from literature, and an evaluated LSTM.

## Acknowledgment

This research was partially supported by MEXT-Japan (Grant No. J17H06100) and NTT Communication Science Laboratories.

## References

[1] H. Sakoe and S. Chiba, "Dynamic programming algorithm optimization for spoken word recognition," *IEEE Trans. Acoustics, Speech, and Sig. Process.*, vol. 26, no. 1, pp. 43–49, 1978.

[2] P. F. Felzenszwalb and R. Zabih, "Dynamic programming and graph algorithms in computer vision," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 4, pp. 721–740, 2011.

[3] H. Ding, G. Trajcevski, P. Scheuermann, X. Wang, and E. Keogh, "Querying and mining of time series data: experimental comparison of representations and distance measures," *Proc. Very Large Data Base Endowment*, vol. 1, no. 2, pp. 1542–1552, 2008.

[4] T. Rakthanmanon, B. Campana, A. Mueen, G. Batista, B. Westover, Q. Zhu, J. Zakaria, and E. Keogh, "Searching and mining trillions of time series subsequences under dynamic time warping," in *Int. Conf. Knowledge Discovery and Data Mining*, 2012, pp. 262–270.

[5] C. Bahlmann, B. Haasdonk, and H. Burkhardt, "Online handwriting recognition with support vector machines-a kernel approach," in *Int. Workshop Frontiers in Handwriting Recognition*, 2002, pp. 49–54.

[6] B. K. Iwana, V. Frinken, K. Riesen, and S. Uchida, "Efficient temporal pattern recognition by means of dissimilarity space embedding with discriminative prototypes," *Pattern Recognition*, vol. 64, pp. 268–276, 2017.

[7] C. Bahlmann and H. Burkhardt, "The writer independent online handwriting recognition system frog on hand and cluster generative statistical dynamic time warping," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, no. 3, pp. 299–310, 2004.

[8] J. Schmidhuber, "Deep learning in neural networks: An overview," *Neural Networks*, vol. 61, pp. 85–117, 2015.

[9] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[10] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification," in *IEEE Int. Conf. Comput. Vision*, 2015, pp. 1026–1034.

[11] D.-A. Clevert, T. Unterthiner, and S. Hochreiter, "Fast and accurate deep network learning by exponential linear units (elus)," *arXiv preprint*, 2015, https://arxiv.org/abs/1511.07289.

[12] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Cognitive modeling*, vol. 5, no. 3, pp. 213–220, 1988.

[13] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[14] I. Sutskever, J. Martens, and G. E. Hinton, "Generating text with recurrent neural networks," in *Int. Conf. Mach. Learning*, 2011, pp. 1017–1024.

[15] A. Graves and N. Jaitly, "Towards end-to-end speech recognition with recurrent neural networks," in *Int. Conf. Mach. Learning*, 2014, pp. 1764–1772.

[16] S. Bai, J. Z. Kolter, and V. Koltun, "An empirical evaluation of generic convolutional and recurrent networks for sequence modeling," *arXiv preprint arXiv:1803.01271*, 2018.

[17] A. Waibel, T. Hanazawa, G. Hinton, K. Shikano, and K. J. Lang, "Phoneme recognition using time-delay neural networks," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 37, no. 3, pp. 328–339, 1989.

[18] R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, and P. Kuksa, "Natural language processing (almost) from scratch," *J. Mach. Learning Research*, vol. 12, no. Aug, pp. 2493–2537, 2011.

[19] Y. Kim, "Convolutional neural networks for sentence classification," in *Conf. Empirical Methods in Natural Language Process.*, 2014, pp. 1746–1751.

[20] N. Razavian and D. Sontag, "Temporal convolutional neural networks for diagnosis from lab tests," *arXiv preprint*, 2015, https://arxiv.org/abs/1511.07938.

[21] J. Yang, M. N. Nguyen, P. P. San, X. Li, and S. Krishnaswamy, "Deep convolutional neural networks on multichannel time series for human activity recognition." in *Int. Joint Conf. Artificial Intell.*, 2015, pp. 3995–4001.

[22] A. v. d. Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior, and K. Kavukcuoglu, "Wavenet: A generative model for raw audio," *arXiv preprint*, 2016, https://arxiv.org/abs/1609.03499.

[23] M. Niepert, M. Ahmed, and K. Kutzkov, "Learning convolutional neural networks for graphs," in *Int. Conf. Mach. Learning*, 2016, pp. 2014–2023.

[24] I. Guyon, L. Schomaker, R. Plamondon, M. Liberman, and S. Janet, "Unipen project of on-line data exchange and recognizer benchmarks," in *Int. Conf. Pattern Recognition*, vol. 2, 1994, pp. 29–33.

[25] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Int. Conf. Learning Representations*, 2015.

[26] E. H. Ratzlaff, "Methods, reports and survey for the comparison of diverse isolated character recognition results on the unipen database," in *Int. Conf. Document Anal. and Recognition*, 2003, pp. 623–628.

[27] B. K. Iwana, V. Frinken, and S. Uchida, "A robust dissimilarity-based neural network for temporal pattern recognition," in *Int. Conf. Frontiers in Handwriting Recognition*, 2016, pp. 265–270.

[28] D. Keysers, T. Deselaers, H. A. Rowley, L.-L. Wang, and V. Carbune, "Multi-language online handwriting recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 6, pp. 1180–1194, 2017.

[29] T. Zahavy, A. Magnani, A. Krishnan, and S. Mannor, "Is a picture worth a thousand words? a deep multi-modal fusion architecture for product classification in e-commerce," *arXiv preprint arXiv:1611.09534*, 2016.

[30] Z. Ahmad and J. Zhang, "Combination of multiple neural networks using data fusion techniques for enhanced nonlinear process modelling," *Comput. & Chemical Engineering*, vol. 30, no. 2, pp. 295–308, 2005.

[31] X. Dai and S. Khorram, "Data fusion using artificial neural networks: a case study on multitemporal change analysis," *Comput., Environ. and Urban Syst.*, vol. 23, no. 1, pp. 19–31, 1999.

[32] F. J. Ordóñez and D. Roggen, "Deep convolutional and lstm recurrent neural networks for multimodal wearable activity recognition," *Sensors*, vol. 16, no. 1, p. 115, 2016.

[33] X. Du, M. El-Khamy, J. Lee, and L. Davis, "Fused dnn: A deep neural network fusion approach to fast and robust pedestrian detection," in *IEEE Winter Conf. App. Comput. Vision*. IEEE, 2017, pp. 953–961.