# Action Recognition with the Augmented MoCap Data using Neural Data Translation

Shih-Yao Lin
mike.lin@ieee.org
Tencent America
Palo Alto, CA, USA

Yen-Yu Lin
yylin@citi.sinica.edu.tw
Academic Sinica
Taipei, Taiwan

**Abstract**

This study aims at generating reliable augmented training data to learn a robust deep model for action recognition. The prior knowledge inferred from few training data is not sufficient to well represent the real data distribution, which makes action recognition quite challenging. Inspired by the recent advances in neural machine translation, we propose a *neural data translation* (NDT) to tackle the aforementioned issue by directly learning the mapping between paired data of the same action class in an end-to-end fashion. The proposed NDT is a sequence-to-sequence generative model. It can be trained with few paired training data, and generates an abundant set of augmented actions with diverse appearance. Specifically, we adopt stochastic pair selection to compile a set of paired training data. Each pair consists of two actions of the same class. One action serves as the input to NDT, while the other acts as the desired output. By learning the mapping between data of the same class, NDT implicitly encodes the intra-class variations so that it can synthesize high-quality actions for augmentation. We evaluated our method on two public datasets, including the Florence3D-action and UCI HAR datasets. The promising results demonstrate that the actions generated by our method effectively improve the performance of action recognition with few examples.

## 1 Introduction

Recognizing human actions has drawn increasing attention for decades due to its wide applicability to many areas such as surveillance, robotics, health care, and human-computer interaction. Recent research efforts, e.g., [8, 10, 11, 14, 15, 18, 19, 22, 23, 27, 32], have successfully applied *deep neural networks* (DNN) to learn and infer human actions from videos. Despite the good performance, learning actions with DNN requires a vast amount of training examples. This requirement may not be satisfied in practice.

Transfer learning such as [20, 25, 31] is one widely adopted solution to learning the neural networks with limited training data. A DNN model is trained in advance with a large dataset in the source domain. By learning the transformation from the source to target domains, the DNN model in the target domain can re-use the parameters from that in the source domain and is fine-tuned with limited training data. However, transfer learning works when data modalities in the source and target domains are the same, e.g., images. In modern real-world applications, data can be captured by various emerging or customized devices, such as accelerometers, gyroscopes, data-gloves, and optical motion-capture systems. Those data

are device-specific. Most transfer learning schemes are inapplicable in this case where cross-modality knowledge transfer is required. In addition, compared with popular modalities such as RGB videos and audio, collecting a large set of such device-specific training data is even more difficult.

*Neural machine translation* (NMT) [2, 6, 9, 28, 33, 34] has been an active research topic. NMT is an end-to-end trainable model for automated text translation. NMT typically contains two main components, namely an encoder network and a decoder network. On the other hand, variational recurrent auto-encoder (VRAE) [5, 7, 8, 17, 24] works on source and target data of the same modality, and learns the mapping from source data to the latent representation. Unlike VRAE, the encoder network in NMT transforms the source sentences into a list of vectors, and the decoder network produces the output sentences for those vectors. Although the structures of the source and output sentences may be diverse, they have similar semantic meaning. NMT can learn the implicit representation within the source and target sentences.

Inspired by the mapping power of NMT, we aim to leverage it to explore the intra-class variations so that the generated actions are good enough for effective data augmentation. We present a *neural data translation* (NDT) model, which is *recurrent neural networks* (RNN)-based auto-encoder, and can perform action to action mapping. Specifically, we adopt a stochastic pair selection scheme, with which a set of paired actions are collected. Each pair contains two actions of the same class. NDT takes one of them as the input, and considers the other the desired output. Learning NDT in this way can encode the implicit structure of the actions and translate actions of the same class. It turns out that high-quality actions are generated by using the learned NDT.

For learning and inferring human actions, we leverage a deep residual bidirectional RNN working with both the original and generated training actions. We test our approach on two public benchmarks for action recognition, including the Florence3D-action and UCI HAR datasets. The experimental results show that the generated actions significantly improve the recognition performance of the learned classifier.

# 2 Our Approach

In this section, a brief review of RNN is firstly given. The proposed neural data translation (NDT)and a deep residual bidirectional RNN classifier are then depicted, respectively.

## 2.1 Recurrent Neural Networks

RNN is a neural network with self-connected recurrent connections for modeling the sequential data. To build a deep RNN, we stack the layers of RNN. A conventional stacked RNN contains a set of hidden state representation $\mathbf{h} = \{h_1, ..., h_t\}$ and an optional output $\mathbf{y}$ which operates on an input sequence $\mathbf{x}$ of an arbitrary length. The output response $\mathbf{h}_t^l$ and $y_t$ can be updated by

$$h_t^l = \sigma(\mathbf{W}_1^l h_t^{l-1} + \mathbf{W}_2^l h_{t-1}^l + b_l) \tag{1}$$

$$h_t^0 := x_t \tag{2}$$

$$y_t = \sigma(\mathbf{V} h_t^L + b_y) \tag{3}$$

where $t \in [1,T]$ and $l \in [1,L]$ represent time steps and layers, respectively. $\sigma$ denotes a non-linear activation function in the hidden layer. It can be the sigmoid function or the hyperbolic tangent function. $\mathbf{W}_1^l$, $\mathbf{W}_2^l$, $\mathbf{V}$ are the mapping weight matrices from the previous hidden layer $h_t^{l-1}$ to the current hidden layer $h_t^l$, from the previous hidden state $h_{t-1}^l$ to the current hidden state $h_t^l$, and from the hidden layer to output $y_t$, respectively. $b_t$ and $b_h$ are the bias vectors.

### 2.1.1 Bidirectional RNN

Bidirectional RNN extends the unidirectional RNN by integrating the forward hidden connection $h_{F,t}^l$ with another backward hidden connection $h_{B,t}^l$. The direction of hidden to hidden state connection of $h_{B,t}^l$ is in opposite temporal order. The output response $\mathbf{h}_{F,t}^l$, $\mathbf{h}_{B,t}^l$ and $y_t$ can be computed by:

$$\begin{aligned} h_{F,t}^l &= \sigma\left(\mathbf{W}_1^l\left(\left[h_{F,t}^{l-1}, h_{B,t}^{l-1}\right]\right) + \mathbf{W}_2^l h_{F,t-1}^l + b_{F,l}\right), \\ h_{B,t}^l &= \sigma\left(\mathbf{W}_1^l\left(\left[h_{F,t}^{l-1}, h_{B,t}^{l-1}\right]\right) + \mathbf{W}_2^l h_{B,t+1}^l + b_{B,l}\right), \text{ and} \\ y_t &= \sigma(\mathbf{V}([h_{F,t}^L, h_{B,t}^L]) + b_y) \end{aligned} \tag{4}$$

### 2.1.2 RNN with Long-Short-Term Memory

Due to the vanishing and exploding gradient effects, RNN has the problem of learning long-range dependencies. To solve it, RNN with *long-short-term memory* (LSTM) [13] has been designed to combat the vanishing and exploding gradient problems, and learns the long-range contextual information of a time series data. Except for the hidden output $h_t$, each LSTM neuron includes an input gate $i_t$, a forget gate $f_t$, an internal memory cell $c_t$, and an output gate $o_t$. With these three gates, the LSTM neuron can choose when to write, read or reset the memory cell at each timestamp. The above scheme allows LSTM to memorize and access information many timesteps ago. Those can be computed by:

$$\begin{aligned} i_t &= \sigma\left(\mathbf{W}_{1,i}^l h_t^{l-1} + \mathbf{W}_{2,i}^l h_{t-1}^l + b_{l,i}\right) \\ f_t &= \sigma\left(\mathbf{W}_{1,f}^l h_t^{l-1} + \mathbf{W}_{2,f}^l h_{t-1}^l + b_{l,f}\right) \\ o_t &= \sigma\left(\mathbf{W}_{1,o}^l h_t^{l-1} + \mathbf{W}_{2,o}^l h_{t-1}^l + b_{l,o}\right) \\ g_t &= \sigma\left(\mathbf{W}_{1,g}^l h_t^{l-1} + \mathbf{W}_{2,g}^l h_{t-1}^l + b_{l,g}\right) \\ c_t &= f_t \odot c_{t-1} + i_t \odot g_t \\ h_t &= o_t \odot \phi\left(c_t\right) \end{aligned} \tag{5}$$

where the operation $\odot$ denotes element-wise multiplication.

## 2.2 Neural Data Translation (NDT) Network

A tiny-scale dataset of $N$ action sequences is given, $D = \{(\mathbf{x}_i, y_i)\}_{i=1}^N$, where each action instance $\mathbf{x}_i$ is temporally normalized and consists of $T_i$ time stamps or frames, $x_{i,t} \in \mathbb{R}^d$, *i.e.*, $\mathbf{x}_i = \{x_{i,1}, x_{i,2}, ..., x_{i,T_i}\}$. $y_i \in \mathcal{Y}$ is its class label. $\mathcal{Y}$ is the class label set. With our stochastic pair selection, $M$ training pairs of the same categories are randomly selected from $\{\mathbf{x}_i, \mathbf{x}_j | \mathbf{x}_i \neq \mathbf{x}_j, y_i = y_j\}$, where $\mathbf{x}_i$ and $\mathbf{x}_j$ present the source action and the target action, respectively. In our pair selection scheme, a source action may be paired with several different target actions.

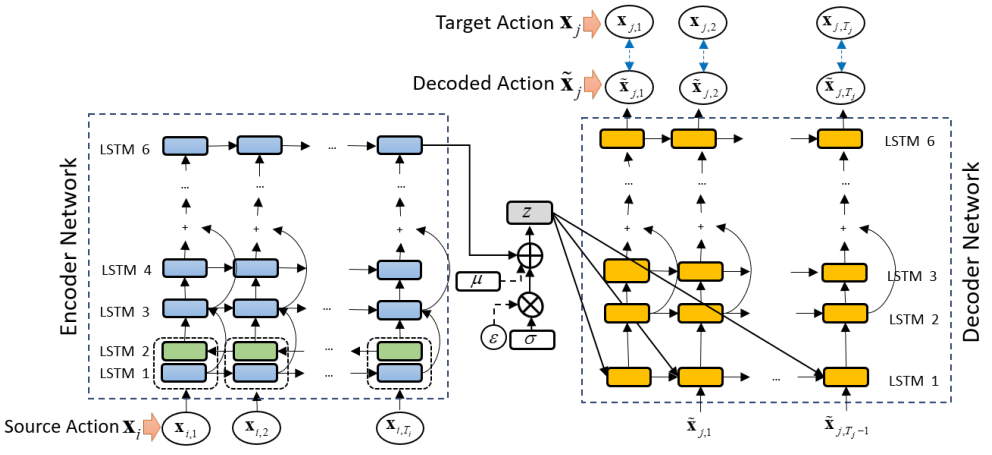


Figure 1: The architecture of our neural data translation (NDT). Our NDT contains two principal components, a encoder network on the left and a decoder network on the right, where $\mathbf{x}_i$, $\tilde{\mathbf{x}}_j$, and $\mathbf{x}_j$ denotes the source action, the decoded output, and the target action, respectively. $T_i$ and $T_j$ expresses the length of source action $\mathbf{x}_{i,t}$ and target action $\mathbf{x}_{j,t}$, respectively. $\mathbf{z}$ is an internal reparameterization vector. $\varepsilon$, and $\mu$ denote the parameter sets of the conditional prior distribution. $\sigma$ presents a noise sampled by a Gaussian distribution.

The common *neural machine translation* (NMT) is an RNN-based sequence-to-sequence learning framework [4, 33], which can translate a sentence from one language to another language and make translated sentence retains the meaning of the original one. Inspired by NMT, our approach aims at learning a translation mapping from an input action sequence $\mathbf{x}_i$ (source action) to another one $\mathbf{x}_j$ (target action). Yet, unlike the general encoder-decoder framework, our proposed *neural data translation* (NDT) is a variational model [34] which maps an source action $\mathbf{x}_i$ to a continuous distribution of latent variables $\mathbf{z}$. The scheme can explicitly model underlying semantics of the action pairs. The conditional probability of $\mathbf{x}_j$ can be formulated as

$$p(\mathbf{x}_j|\mathbf{x}_i) = \int_{\mathbf{z}} p(\mathbf{x}_j|\mathbf{z},\mathbf{x}_i)\, p(\mathbf{z}|\mathbf{x}_i)\, d_{\mathbf{z}} \tag{6}$$

Follow the derivation of [8], the variational lower bound of our NDT can be formulated as follows:

$$\mathcal{L}(\theta,\phi;\mathbf{x}_j,\mathbf{x}_i) = -D_{KL}\left(q_\phi(\mathbf{z}|\mathbf{x}_i,\mathbf{x}_j)\,||\,p_\theta(\mathbf{z}|\mathbf{x}_i)\right) + \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x}_i,\mathbf{x}_j)}\left[\log p_\theta(\mathbf{x}_j|\mathbf{z},\mathbf{x}_i)\right], \tag{7}$$

where $D_{KL}(p|q)$ denotes the Kullback-Leibler divergence between distribution $p$ and $q$. $p_\theta(\mathbf{z}|\mathbf{x}_i)$ is the prior model, $q_\phi(\mathbf{z}|\mathbf{x}_i,\mathbf{x}_j)$ is the posterior approximator, and $p_\theta(\mathbf{x}_j|\mathbf{z},\mathbf{x}_i)$ is the decoder with the guidance from latent variable $\mathbf{z}$.

According to Eq. 7, our NDT can be divided into two principal components, a encoder network, a decoder network. Each of them can be modeled by multi-stacked LSTMs, as shown in Fig. 1. We leverage eight encoder layers and eight decoder layers. In the bottom of encoder layer, we adopt a bidirectional-LSTM. The *encoder network* learns the internal

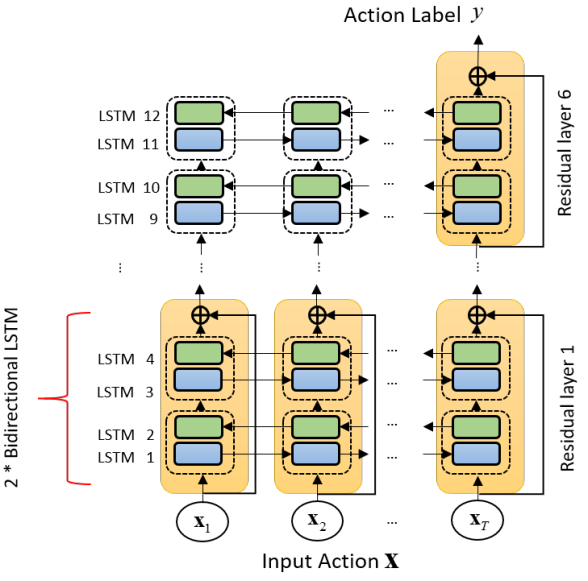


Figure 2: Architecture of the deep residual bidirectional-LSTM. $\mathbf{x}$ denotes the input action and $y$ expresses the output action label. In our model, we utilize a residual connections between stacked bidirectional-LSTM to forward the underlying information to the upper layer. Our model contains six residual-layers, where each residual layer has two bidirectional-LSTM.

distributed representation of the source actions. The *decoder network* models $p_\theta(\mathbf{x}_j|\mathbf{z},\mathbf{x}_i)$., and the *variational inference* models $p_\theta(\mathbf{z}|\mathbf{x}_i)$ and $q_\theta(\mathbf{z}|\mathbf{x}_i,\mathbf{x}_j)$. To generate the augmented training actions, we adopt the trained parameters of the decoder network to achieve it.

### 2.3 Deep Residual Bidirectional-LSTM Classifier

Deeper recurrent neural networks are generally difficult to train. To avoid the gradient vanishing problem, we leverage a residual bidirectional-LSTM [12] to learn and infer human actions. The residual connections between stacked LSTM cells act as highways for gradients, which can forward underlying information directly to the upper layer. Fig. 2 demonstrates the architecture of our residual bidirectional-LSTM model. Our classifier is built by using stacked bidirectional LSTM cells and residual LSTM cells for every stacked layer. Each residual layer has two bidirectional LSTMs, i.e., four unidirectional LSTM cells.

## 3 Experimental Results

This section first presents the setting of the conducted experiments, including the used datasets and evaluation metrics. The experimental results and the analysis are then described.

## 3.1 Datasets for Evaluation

Out method is evaluated on two public datasets, including Florence3D Action [26] and UCI HAR datasets [1].

### 3.1.1 The Florence3D Action dataset

This dataset is a small-scale dataset with only 215 action instances, which were captured by a Microsoft Kinect sensor. This dataset contains 9 daily actions performed by 10 subjects. The action classes are *wave, drink from a bottle, answer phone, clap, tight lace, sit down, stand up, read watch,* and *bow*. The 3D locations of 15 body joints are provided. Difficulties, such as large intra-class variations and high inter-class similarity, are present and make this dataset challenging.

### 3.1.2 The UCI HAR dataset

This dataset include 6 actions performed by 30 subjects. It contains 10,299 actions. The action categories contain *walking, walking upstairs, walking downstairs, sitting, standing,* and *laying*. Each subject performed and wore a waist-mounted smartphone which is embedded an accelerometer and a gyroscope sensors on their waists. Each training example has 128 frames. Each frame is with the three signals, each of which at each time stamp is in form of a 3-dimensional vector.

## 3.2 Evaluation Metrics

For the Florence3D action dataset, each action is represented by the absolute 3D body joint locations in the skeletal stream. Each action example consists of $T = 35$ skeleton frames which are uniformly sampled from each action. The normalization process in [29] is adopted for making the skeletons invariant to the absolute location of actors. We adopt cross-subject-testing [26], where half of the subjects are used for training and the rest are used for testing. We then switch their roles and report the average performance. For the UCI-HAR dataset, we adopt cross-subject-testing where 70% of the subjects are randomly selected and used for training, while the rest subjects are used for testing.

## 3.3 Setting of Data Augmentation

For Florence3D action dataset, we apply two common data augmentation schemes for increasing the diversity of human poses. We generate three types of augmented data, which will be introduced in the following.

First, we mirror the human pose horizontally for each original action data to obtain extra 215 mirrored action examples, and we term them *mirrored actions*. The second type of augmented data are constructed based on the motion variation of human actions. We randomly add Gaussian noise to each body joint locations of the original action examples and the mirrored actions to obtain a total of $215 \times 2 = 430$ synthesis action instances. The original action and mirrored action with added Gaussian noise are named them *noisy actions* and *noisy mirrored actions*. By leveraging our basic data augmentation, the original dataset has then been augmented with 860 action examples. More examples of our augmented action examples with our basic data augmentation scheme can be found in our supplementary video: https://sites.google.com/view/action-recognition-ndt/.

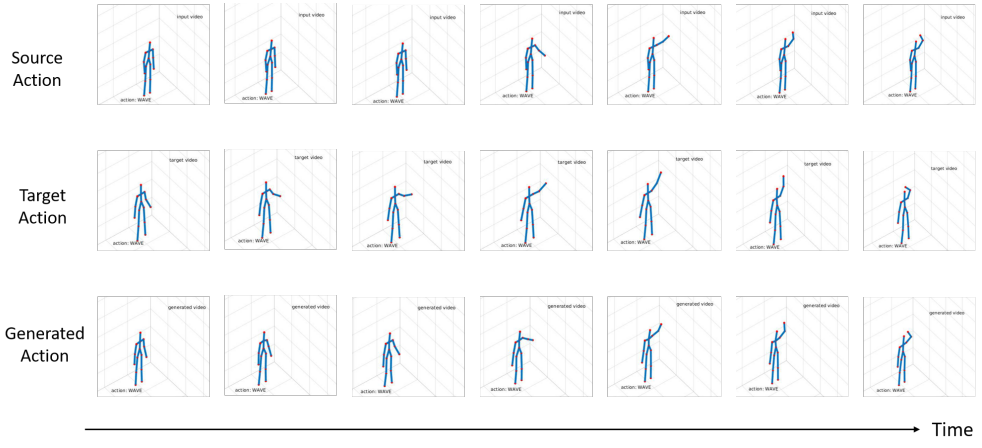


Figure 3: The examples of the generated *waving-hand* action by our NDT, and its source and target actions. The first and second rows illustrate the source and target actions within a selected pair. The bottom of the row shows the generated action by our NDT. The action style and speed of our generated action is certainly different to the source and target actions. More importantly, our generated action still belongs to the *waving-hand* action category, same as the source and target actions.

Moreover, we generated 215 training examples by using our NDT. Figure 3 illustrates some examples of the generated *waving-hand* actions by our NDT and its source and target action. In this case, we take the source action as input to our NDT. The generated actions can then be obtained by our NDT. As shown in Figure 3, the generated action is like neither the source nor target action. It performs different motion styles and speeds to the source and target actions. More importantly, the generated action still belongs to the *waving-hand* action category, the same as the source and target actions. More experimental results can be found in our supplementary videos.

For the UCI HAR dataset, we generated 1,000 augmented training examples by using our NDT. The training examples generated by our NDT are termed *NDT actions*.

## 3.4 Evaluation Results

For the Florence3D action dataset, we chose seven existing approaches for comparison with the proposed NDT, including, Multi-Part Bag-of-Poses [26], Elastic Functional Coding [2], Skeletons Lie group [30], ConvESN-MSMC [22], Tensor Representations [16], the approach by Vemulapalli *et al.* [29], and approach by Luvizon *et al.* [21]. Moreover, we implemented variants of LSTM-based methods including Two-recurrent-layers RNN (DeepStackedLSTM), Two-recurrent-layers bidirectional RNN (DeepStackedBirLSTM), and Residual bidirectional LSTM (ResDeepStackedBirLSTM).

The recognition accuracy of all methods on the Florence3D action dataset is shown in Table 1. Our ResDeepStackedBirLSTM achieves the recognition rate of 95.0%, and performs favorably against the other RNN-based methods we implemented. We think the reason is that the network can train deeper by using residual connection. In addition, the state-of-the-art

Table 1: Results on the Florence3D-Action dataset.

| Method | Accuracy (%) |
|---|---|
| Multi-Part Bag-of-Poses [26] | 82.0 |
| Elastic Functional Coding [2] | 89.6 |
| Skeletons Lie group [30] | 90.7 |
| ConvESN-MSMC [22] | 91.7 |
| Vemulapalli *et al.* [29] | 91.4 |
| Luvizon *et al.* [21] | 94.3 |
| Tensor Representations [16] | **95.2** |
| Our DeepStackedLSTM | 91.2 |
| Our DeepStackedBirLSTM | 93.3 |
| Our ResDeepStackedBirLSTM | 95.0 |
| Our ResDeepBirLSTMs with *noisy actions* | **95.2** |
| Our ResDeepBirLSTMs with *mirrored actions* | **96.2** |
| Our ResDeepBirLSTMs with *noisy mirror actions* | **96.5** |
| Our ResDeepBirLSTMs with *NDT actions* | **96.4** |
| Our ResDeepBirLSTMs with *mirrored + NDT actions* | **96.8** |

Table 2: Results on the UCI HAR database.

| Method | Accuracy (%) |
|---|---|
| MC-SVM [1] | 89.0 |
| MC-HF-SVM [1] | 89.3 |
| DeepStackedLSTM | 91.0 |
| DeepBirLSTM | 92.3 |
| ResDeepBirLSTM | 94.0 |
| ResDeepBirLSTMs with *NDT actions* | **95.3** |

method [16] reaches the recognition rate of 95.2%, which still outperforms all the existing methods. As shown in Table 1, we found that our ResDeepStackedBirLSTM trained on original training examples with *NDT actions* achieve recognition rate of 96.4%. The improvement in recognition rate of $1.4\%(=96.4\%-95.0\%)$ is gained when comparing with the case where only the original data are used.

Besides, our ResDeepStackedBirLSTM trained with generated mirrored actions achieves the recognition rate of 96.2%. Yet, our model trained with the noisy actions only gains minor improvement. We think the reason is that the human motion generated in the NDT actions and the mirrored actions have rich variations, and the noisy actions only provide restricted variation of the original training data. Our model trained with the NDT and mirrored actions achieves the recognition rate of 96.7%. Yet, the experimental results by training our classifier with the NDT and noisy mirrored actions dose not lead to a performance improvement.

For the UCI HAR dataset, we chose five existing methods including multiclass SVM (MC-SVM [1]), multiclass Hardware-Friendly SVM (MC-HF-SVM [1]), DeepStackedLSTM, DeepBirLSTM, and ResDeepBirLSTM. The recognition results of all the evaluated approaches on the UCI HAR dataset are shown in Table 2. The results indicate that our ResDeepBirLSTM gains the recognition rate of 94.0% which is more favorable than those by

all the competing methods. On the other hand, ResDeepBirLSTM trained with *NDT actions* improves the recognition rate by $1.3\%(= 95.3\% - 94.0\%)$. The experimental results show that the classifier trained with our generated data can raise the recognition performance.

# 4 Conclusions

In this study, we present a sequence-to-sequence generative model, named neural data translation (NDT), which explores the intra-class variations and discovers the intrinsic data structures so that it can generate high-quality training data from few training action examples. We tested our method on two public action datasets. The experimental results demonstrate that the classifier for human action recognition can be greatly enhanced and lead to significant performance gains by training it with both the original actions and those generated by our method. For future work, we plan to enhance and apply the proposed method to more computer vision applications where collecting training data is expensive.

# 5 Acknowlegement

This work was supported in part by the Ministry of Science and Technology, Taiwanli under grant MOST105-2221-E-001-030-MY2.

# References

[1] Davide Anguita, Alessandro Ghio, Luca Oneto, Xavier Parra, and Jorge Luis Reyes-Ortiz. A public domain dataset for human activity recognition using smartphones. In *ESANN*, 2013.

[2] Rushil Anirudh, Pavan Turaga, Jingyong Su, and Anuj Srivastava. Elastic functional coding of human actions: From vector-fields to latent variables. In *CVPR*, pages 3147–3155, 2015.

[3] Seungryul Baek, Zhiyuan Shi, Masato Kawade, and Tae-Kyun Kim. Kinematic-layout-aware random forests for depth-based action recognition. In *BMVC*, 2016.

[4] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. In *ICLR*, 2015.

[5] Mathias Berglund, Tapani Raiko, Mikko Honkala, Leo Kärkkäinen, Akos Vetek, and Juha T Karhunen. Bidirectional recurrent neural networks as generative models. In *NIPS*, pages 856–864, 2015.

[6] Samuel R Bowman, Luke Vilnis, Oriol Vinyals, Andrew M Dai, Rafal Jozefowicz, and Samy Bengio. Generating sentences from a continuous space. *arXiv preprint arXiv:1511.06349*, 2015.

[7] Junyoung Chung, Kyle Kastner, Laurent Dinh, Kratarth Goel, Aaron C Courville, and Yoshua Bengio. A recurrent latent variable model for sequential data. In *NIPS*, pages 2980–2988, 2015.

[8] Otto Fabius and Joost R van Amersfoort. Variational recurrent auto-encoders. *CoRR abs/1412.6581*, 2014.

[9] Marzieh Fadaee, Arianna Bisazza, and Christof Monz. Data augmentation for low-resource neural machine translation. In *ACL*, pages 567–573, 2017.

[10] Jiyang Gao, Zhenheng Yang, and Ram Nevatia. Cascaded boundary regression for temporal action detection. In *BMVC*, 2017.

[11] Ankur Gupta, Alireza Shafaei, James J Little, and Robert J Woodham. Unlabelled 3d motion examples improve cross-view action recognition. In *BMVC*, 2014.

[12] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016.

[13] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[14] Yueyu Hu, Chunhui Liu, Yanghao Li, Sijie Song, and Jiaying Liu. Temporal perceptive network for skeleton-based action recognition. In *BMVC*, 2017.

[15] Yu Kong, Zhiqiang Tao, and Yun Fu. Deep sequential context networks for action prediction. In *CVPR*, pages 1473–1481, 2017.

[16] Piotr Koniusz, Anoop Cherian, and Fatih Porikli. Tensor representations via kernel linearization for action recognition from 3d skeletons. In *ECCV*, pages 37–53, 2016.

[17] Jiwei Li, Minh-Thang Luong, and Dan Jurafsky. A hierarchical neural autoencoder for paragraphs and documents. *arXiv preprint arXiv:1506.01057*, 2015.

[18] Shih-Yao Lin, Yen-Yu Lin, Chu-Song Chen, and Yi-Ping Hung. Learning and inferring human actions with temporal pyramid features based on conditional random fields. In *ICASSP*, pages 2617–2621, 2017.

[19] Shih-Yao Lin, Yen-Yu Lin, Chu-Song Chen, and Yi-Ping Hung. Recognizing human actions with outlier frames by observation filtering and completion. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, 13(3): 28, 2017.

[20] Fang Liu, Xiangmin Xu, Shuoyang Qiu, Chunmei Qing, and Dacheng Tao. Simple to complex transfer learning for action recognition. *TIP*, 25(2):949–960, 2016.

[21] Diogo C. Luvizon, Hedi Tabia, and David Picard. Learning features combination for human action recognition from skeleton sequences. *Pattern Recognition Letters*, 2017. doi: http://dx.doi.org/10.1016/j.patrec.2017.02.001.

[22] Qianli Ma, Lifeng Shen, Enhuan Chen, Shuai Tian, Jiabing Wang, and Garrison W Cottrell. Walking walking walking: Action recognition from action echoes. In *IJCAI*, pages 2627–2633, 2017.

[23] Pascal Mettes, Cees GM Snoek, and Shih-Fu Chang. Localizing actions from video labels and pseudo-annotations. In *BMVC*, 2017.

[24] Yunchen Pu, Zhe Gan, Ricardo Henao, Xin Yuan, Chunyuan Li, Andrew Stevens, and Lawrence Carin. Variational autoencoder for deep learning of images, labels and captions. In *NIPS*, pages 2352–2360, 2016.

[25] Hossein Rahmani, Ajmal Mian, and Mubarak Shah. Learning a deep model for human action recognition from novel viewpoints. *TPAMI*, 2017.

[26] Lorenzo Seidenari, Vincenzo Varano, Stefano Berretti, Alberto Bimbo, and Pietro Pala. Recognizing actions from depth cameras as weakly aligned multi-part bag-of-poses. In *CVPRW*, pages 479–485, 2013.

[27] Amir Shahroudy, Jun Liu, Tian-Tsong Ng, and Gang Wang. Ntu rgb+ d: A large scale dataset for 3d human activity analysis. In *CVPR*, pages 1010–1019, 2016.

[28] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Sequence to sequence learning with neural networks. In *NIPS*, pages 3104–3112, 2014.

[29] Raviteja Vemulapalli and Rama Chellapa. Rolling rotations for recognizing human actions from 3d skeletal data. In *CVPR*, pages 4471–4479, 2016.

[30] Raviteja Vemulapalli, Felipe Arrate, and Rama Chellappa. Human action recognition by representing 3d skeletons as points in a lie group. In *CVPR*, pages 588–595, 2014.

[31] Yu-Xiong Wang and Martial Hebert. Learning to learn: Model regression networks for easy small sample learning. In *ECCV*, pages 616–634, 2016.

[32] Junwu Weng, Chaoqun Weng, and Junsong Yuan. Spatio-temporal naive-bayes nearest-neighbor (st-nbnn) for skeleton-based action recognition. In *CVPR*, pages 4171–4180, 2017.

[33] Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, et al. Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*, 2016.

[34] Biao Zhang, Deyi Xiong, Jinsong Su, Hong Duan, and Min Zhang. Variational neural machine translation. *arXiv preprint arXiv:1605.07869*, 2016.