

CONTINUAL PALMPRINT RECOGNITION WITHOUT FORGETTING

Xuefeng Du¹, Dexing Zhong¹, Huikai Shao¹

¹School of Electronic and Information Engineering, Xi'an Jiaotong University,
28 West Xianning Road, Xi'an, Shaanxi 710049, P. R. China

ABSTRACT

As a promising topic of biometrics, palmprint recognition helps to effectively verify a person's identity, which is suitable for building a security system. Recent progress has achieved high recognition accuracy in different benchmark datasets due to deep learning. However, these applications are almost implemented in one dataset with iterative training epochs to help neural network generalize. When applied practically where many new users' palmprints registered in sequence, deep learning-based recognition systems cannot avoid the problem of catastrophic forgetting. In this paper, we propose a continual learning framework based on reinforcement learning to dynamically expand the neural network when facing newly registered palmprints without costly retraining or fine-tuning. Experiments on different datasets demonstrate the high adaptability of our model that is promising for solving the forgetting attack of every biometric system.

Index Terms— Biometrics, Palmprint recognition, Continual learning, Information security

1. INTRODUCTION

Biometrics, which uses human's physiological and behavioral characteristics for identification, has been researched extensively. Compared to some popular methods, such as face [1], and fingerprint [2], palmprint recognition is more reliable because it focuses on larger feature area with more discriminative features, such as line, ridge and texture. Meanwhile, palmprint pattern cannot change unless under surgeries and is highly user friendly.

Recently, researchers have exploited many machine learning techniques to represent palmprints, such as PCA [3], LBP [4], *etc.* Then, classifiers are trained using registered palmprints for recognizing unseen images. Due to the progress in deep learning, the feature engineering process becomes simpler and more informative which leads to increasing accuracy on many individual recognition tasks [5-10]. Recently, we applied Siamese network to compare different palmprint images, which takes over 5000 epochs to converge on large-scale dataset with the accuracy close to 100%. In practical scenarios, a palmprint recognition system should adapt to different data quickly because new identities

are continuously registered. Usually, the data distribution changes frequently due to variations in illumination, noise and hand poses, *etc.* However, current recognition methods rely on iterative training on a single domain and it struggles when the data distribution changes over the course of learning. When trained on new task, neural networks forget most knowledge obtained from previous tasks, which is called "catastrophic forgetting" [11]. One alternative solution is fine tuning which leads to degenerate performance in both previous tasks and new task due to change in data distribution.

In this paper, to build a secure palmprint recognition system, we propose a continual learning framework to help the recognition system adapt to new tasks quickly without catastrophic forgetting. Firstly, we specify the continual palmprint recognition scheme into two types. The first is incremental testing where different tasks contain images collected from the same device but with non-overlapping classes. The second type is cross domain recognition where different tasks contain palmprints from different devices with different label space as well. For each task coming in sequence, we dynamically expand the neural network by deciding the optimal filters or nodes to add for each layer according to [12]. We apply reinforcement learning to solve this combinatorial optimization problem with self-designed reward functions. Besides, a Gated Recurrent Unit (GRU) module is adopted to generate the probabilities of actions to take. To speed up the convergence, we apply the Actor-Critic strategy to train the decision maker which considers both the model complexity and recognition performance. When trained on later tasks, the neural network does not see previous palmprint data and only optimizes newly added structures, which decreases the model complexity to a large extent.

The overall architecture is shown in Fig.1.

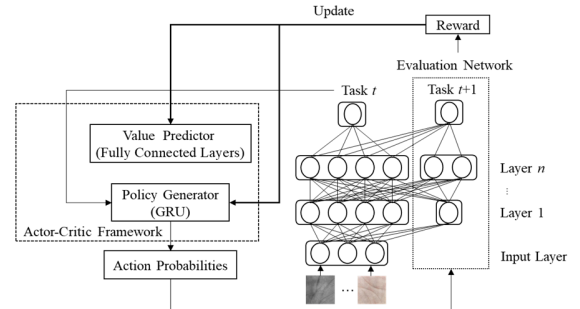


Fig.1 The flowchart of our continual recognition framework

The rest of the paper is organized as follows: Section 2 reviews related works. Section 3 describes our proposed continual palmprint recognition framework. Section 4 explains the experiments and results. Section 5 concludes the paper.

2. RELATED WORKS

Palmprint recognition has been researched over many years. Existing methods typically define local or global based features after Region of Interest (ROI) cropping [13]. Then, some classifiers are applied for either the verification or identification [14]. Minaee [15] *et.al.* first proposed an image representation method called scattering network. Afterwards, many researchers have introduced deep learning into palmprint recognition [16-20]. However, most of these algorithms were implemented by training Convolutional Neural Network (CNN) repeatedly, so the generalization ability is limited. When new data comes, additional retraining on the whole dataset or fine-tuning is necessary to avoid catastrophic forgetting. Sometimes, the retraining takes thousands of epochs to converge on a large dataset while fine-tuning will cause the semantic drift once the tasks trained later is quite different from the previous ones.

Continual learning is a learning framework where the model learns from a sequence of tasks while transferring knowledge obtained from earlier tasks to later ones. Recent works always focus on training efficiently and fast knowledge transfer. To prevent the semantic drift, some researchers introduced regularization to parameters without changing the model structure which aims to avoid semantic drift when training on later tasks [21, 22]. However, the regularizer prevents learning discriminative knowledge from new tasks. Fernando *et.al.* [23] proposed PathNet, where an agent discovers which parts of the network to reuse for new tasks via genetic algorithm. However, these methods do not dynamically change the network structure, therefore the space complexity is large. Another type is dynamically expanding the network while keeping the old parts fixed. Rusu *et.al.* [24] proposed Progressive Neural Networks to expand the network with fixed nodes which led to a large-capacity network structure. Dynamically Expandable Network (DEN) [25] introduced group sparsity regularization term to alleviate this problem, but too many hyperparameters are very hard to adjust for an optimal network structure.

In our paper, we dynamically change the structure in every layer by making decisions using reinforcement learning by referring to [12]. We don't incorporate many thresholding values, but consider the model complexity and test accuracy in reward functions, so that it avoids catastrophic forgetting with fewer training parameters at the same time.

3. METHOD

In Fig.1, the continual palmprint recognition system consists of four parts, which are initial recognition neural network, expanded evaluation network trained on later tasks, a policy

generator which is used to generate the probabilities of each action been taken for deciding the expanding scheme in each layer. We use a two-layer GRU with the number of actions to take as the number of timesteps. The value estimator is used to approximate the reward of an expanding scheme. We apply a fully connected layer as this value estimator.

3.1. Policy Generator

In Fig.1, after the training on task t , we use the policy generator to obtain the optimal filters and neurons to add for task $t+1$. In order to prevent semantic drift, we keep the parameters learned in task t fixed according to [25] and only apply gradient descent on the newly added part. When we evaluate the recognition accuracy of task t , we don't consider the structure introduced in later tasks.

If we assume that the neural network has n layers and the maximum number of filters or neurons to be added is M . For an optimal solution to the network structure, directly traversing n layers with M possible expanding scheme at each layer, we obtain M^n solutions which are infeasible to compare for large-capacity networks. Therefore, we refer to recurrent neural network and regard the expanding decision as a string whose length is the number of layers to expand. Since there is a recurrent relationship between neighboring layers, at the first timestep, we input a random probability vector whose size is M to the two-layer GRU. For task t at layer k , we calculate the output $P_t^k \in \mathbb{R}^M$, which is the probabilities of sampling each action for layer k in the SoftMax form. We then regard the output P_t^k as the input of layer $k+1$ and iteratively obtain the expanding decision string. Therefore, a sequence of actions (a policy) follows such a probability:

$$\pi(Act(1...n) | S, \theta_p) = \prod_{k=1}^n P_t^k \quad (1)$$

Where θ_p refers to the parameters of the policy generator. S is the probability vector of each action and is the 'state' defined in reinforcement learning. $Act(1...n)$ denotes the action sequence.

3.2. Evaluation Network

Suppose we have T recognition tasks coming in sequence denoted as $Task(t) = \{x_i, y_i\}_{i=1}^{L(t)}, t=1...T$, which is split into training set $\{x_i, y_i\}_{i=1}^{Q(t)}$, validation set $\{x_i, y_i\}_{i=1}^{M(t)}$ and test set $\{x_i, y_i\}_{i=1}^{N(t)}$ where $L(t), Q(t), M(t), N(t)$ refer to the number of palmprint images in each dataset. For the first task, we follow the common supervised learning procedure and train an initial neural network by optimizing the objective function:

$$\min_{W_1} Loss_1(W_1, \{x_i, y_i\}_{i=1}^{Q(t)}) \quad (2)$$

Where W_1 is the parameters of the initial model. Therefore, when we train our model at task t , the expanding policy is generated by the policy generator and we only need to update the parameters $W_t \setminus W_{t-1}$, which is the newly added filters and nodes. We use stochastic gradient descent to update the sub-network and define α as the learning rate.

$$W_t \setminus W_{t-1} = W_t \setminus W_{t-1} - \alpha \nabla_{W_t \setminus W_{t-1}} Loss_t \quad (3)$$

After the specified epochs under one expanding scheme $Act(1..n)$, we test the recognition performance in validation set and testing set whose accuracy is then averaged as $A(t, Act(1..n))$. For every task, we define the maximum trials to generate different expanding policies and we return the best parameters which achieve the highest reward (defined in the next section) and store them as the initial parameters for task $t+1$.

3.3. Reward Function

In order to update the policy generator and value predictor, we design the reward function for each specific action sequence $Act(1..n)$ which takes both the model complexity and the recognition performance into consideration. To avoid a complicated model, we sum up the number of the filters or nodes to add in each layer as the model complexity which is denoted as $C(t, Act(1..n))$. Therefore, the reward of task t is defined as

$$R(t) = A(t, Act(1..n)) - \eta C(t, Act(1..n)) \quad (4)$$

Where η is the offset parameter balancing the two terms.

3.3. Reinforced Training

Since the reward design in (4) is not differentiable, we apply policy gradient to update the Actor-Critic framework.

For every task t , since we obtain the reward based on Equation (4), the policy generator aims to find the right expanding scheme which generates the maximum reward. In the Actor-Critic framework, we introduce a value predictor to approximate the actual reward of an action sequence, which is parametrized by θ_v . We define the estimated reward as $V(S(t), \theta_v)$. According to regular reinforcement learning [26, 27], the gradient of Actor-Critic framework is:

$$\begin{aligned} \nabla_{\theta_p} R(\theta_p) = E \left[\sum_{Act(1..n)} \pi(Act(1..n) | S(t), \theta_p) (R(S(t), Act(1..n)) \right. \\ \left. - V(S(t), \theta_v)) \frac{\nabla_{\theta_p} \pi(Act(1..n) | S(t), \theta_p)}{\pi(Act(1..n) | S(t), \theta_p)} \right] \end{aligned} \quad (5)$$

Following the Monto-Carlo estimation method, Equation (5) in the batch mode can be approximated as:

$$\frac{1}{N} \sum_{batch=1}^N \nabla_{\theta_p} \log \pi(Act(1..n) | S(t), \theta_p) \times (R(S(t), Act(1..n)) - V(S(t), \theta_v)) \quad (6)$$

For the value predictor, the loss function and gradient are

$$Loss = \frac{1}{N} \sum_{batch=1}^N (V(S(t), \theta_v) - R(S(t), Act(1..n)))^2 \quad (7)$$

$$\nabla_{\theta_v} Loss = \frac{2}{N} \sum_{batch=1}^N (V(S(t), \theta_v) - R(S(t), Act(1..n))) \frac{\partial V(S(t), \theta_v)}{\partial \theta_v} \quad (8)$$

Therefore, the policy generator chooses actions based on probabilities and modifies the probability of the chosen action based on value predictor's score. So they complement and optimize each other in an interactive way.

4. EXPERIMENTS AND RESULTS

4.1 Database

Firstly, we describe the datasets used for continual palmprint recognition, which involves the incremental testing in a single domain, and cross domain palmprint recognition.

4.1.1 PolyU Multispectral Database

PolyU multispectral dataset contains four spectral bands, *i.e.* blue (PA), red (PR), green (PG) and NIR (PN). 6000 grayscale images from 500 palms are collected for each band [28], including 195 males and 55 females. For each palm, 12 images are collected. The images are oriented and cropped to form ROI which are 128×128 pixels.

4.1.2 XJTU Uncontrolled Dataset

To simulate the practical image acquisition, we set up four palmprint datasets where the users can freely move their hands. The XJTU-A (XA) image dataset is built in the lab with proper illumination and low noise level. We use CMOS camera to collect 57 people's grayscale images of both left and right palms. For each palm, we collect 10 images. XJTU-B (XB) is constructed outdoors so the quality of the images is lower. We collect 1000 RGB images from 100 different people. The images in dataset XJTU-C (XC) and XJTU-D (XD) are collected by iPhone6s with and without flash light in the RGB form. The number of images is 1960 and 1970 from 98 people.

4.1.3 Tongji Palmprint Dataset

In Tongji (T) palmprint dataset, images were collected from 300 volunteers, including 192 males and 108 females in two sessions. In total, the database contains 12,000 images from 600 different palms [29]. The exemplary images and their ROIs are demonstrated in Fig.2.

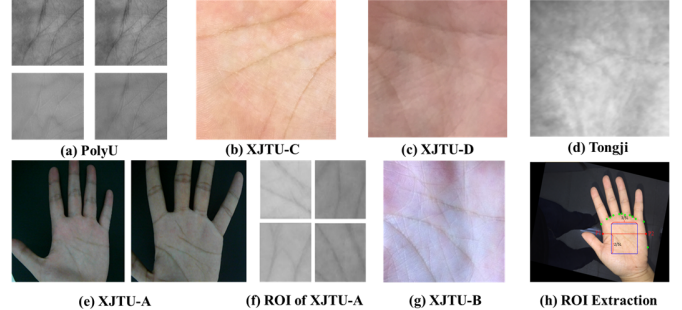


Fig.2 Different datasets used in our experiments. Note that (h) denotes the ROI cropping method. We select two points on the edge of palm (the red point in (h)) as the reference points. A square area, whose side length was three-fifths of the length between the two points, was extracted as ROI.

4.2 Implementation Details

For the initial model, we apply two convolutional layers with the shape of filters to be 3×3 and number of filters to be 16 and 32. Four max-pooling layers which down-sample the feature size by a half are followed. Then, three fully connected layers are stacked with the initial shape to be 2304×512 , 512×128 , $128 \times 50/10$. The offset parameter in (4) is 0.001. The learning rate of recognition model is $1e-4$ while the learning rate of the policy generator and value estimator is 0.001 and 0.005. We set batch size to 32. We set training epochs to be 150 for every task in sequence.

4.2 Incremental Testing

First, we assume the number of tasks in sequence to be 10 with 50 classes in every task for PolyU and Tongji dataset and 10 classes for XJTU datasets. For incremental testing, we introduce a new set of classes in every coming task, but these images are from the same dataset and collected under similar condition. For every class, we resize images to 128×128 pixels. We have in total 12 images in one class. 6 of them are used for training. The half of remaining 6 images are used for validation and others are used for testing. For XJTU datasets, we select two validation images to become testing images so that the total number of images in one class equals 12. For each task, 300 images are for training and 150 images are used for evaluation and testing in Tongji and PolyU dataset with corresponding 60 and 30 images for XJTU datasets. We train all 10 tasks one by one, once one task has been trained for predefined epochs, it will not be observed later. We specify the size of the search space to be 30 for both CNN layers and fully connected layers. We report the average accuracy and the average number of parameters to train over all 10 tasks for 9 datasets. For comparison, we directly apply supervised models trained on 10 tasks and report average accuracy for 9 different datasets. Besides, we also fine-tune the model on every new task from the first task and report the average accuracy on current task and on previous tasks. The overall result of the incremental testing is shown in Table.1.

Table 1 Comparative Experiment Results

Dataset	Supervised training	Fine-tuning		Continual Recognition
		Previous	Current	
PB	99.3%	82.2%	96.6%	97.9%
PR	99.1%	83.6%	98.9%	98.6%
PG	99.0%	84.9%	94.3%	98.7%
PN	99.8%	79.1%	95.6%	98.1%
XA	95.9%	71.5%	93.0%	95.8%
XB	90.7%	69.2%	86.7%	88.7%
XC	96.5%	74.2%	96.5%	95.6%
XD	96.8%	72.3%	95.9%	96.6%
T	94.6%	80.9%	93.7%	94.9%

From Table.1, continual palmprint recognition successfully avoids catastrophic forgetting compared to the performance of fine-tuning in 9 different datasets, whose accuracy on previous tasks drops down over 20% compared to the supervised scenario. When we fine-tune the previous network with new data, the performance on the current task is near to optimal condition, but is slight lower than the performance of continual recognition. The reason is that it aims to change the whole parameter space which is hard to adapt on new tasks. Meanwhile, fine-tuning on later tasks causes semantic drift so the testing on the previous tasks fail. Besides, the recognition performance in XJTU datasets are weaker than that in the benchmark dataset which is caused by the intra-class environmental variations.

In order to compare the model complexity, we plot $C(t, Act(1..n))$ and the average parameters to train for each dataset in Fig. 3 (a). Therefore, our model decreases trainable parameters by a half without influencing the accuracy. The final average expanding scheme is specified in Fig.3 (b).

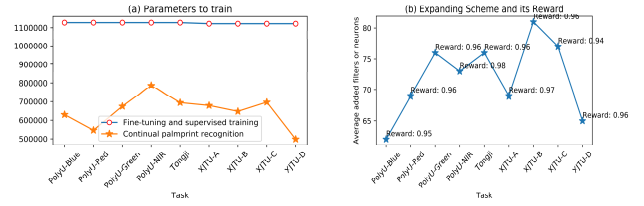


Fig.3 Model complexity and expanding scheme

4.2 Cross Domain Continual Palmprint Recognition

Similar to the setting in 4.1, we set up 9 different tasks where each task contains images from 50 classes in a unique dataset. All the other settings are the same as that of incremental testing. The neural network is required to adapt to data both from different classes but from different domains. We test five different sequential combinations of the datasets, which is randomly decided.

The comparative results are shown in Table.2.

Table 2 Comparative Experiment Results

Combination	Supervised training	Fine-tuning		Continual Recognition
		Previous	Current	
1	95.5%	79.1%	95.0%	94.1%
2	96.2%	81.5%	94.6%	92.8%
3	94.4%	75.0%	91.1%	90.9%
4	93.6%	79.6%	94.4%	93.9%
5	94.7%	80.1%	91.5%	92.5%

From Table.2, continual recognition in cross domain datasets has a weaker performance than testing on the single domain. Compared to fine-tuning on current task, our proposed method is also slightly weaker because of the variations in different datasets. Meanwhile, the accuracy of supervised training decreases as well because of the averaging on all 9 tasks. However, our model can still avoid catastrophic forgetting and adapt quickly on cross-domain palmprint recognition tasks, which is very promising. The average model complexity and expanding scheme is shown in Fig.4.

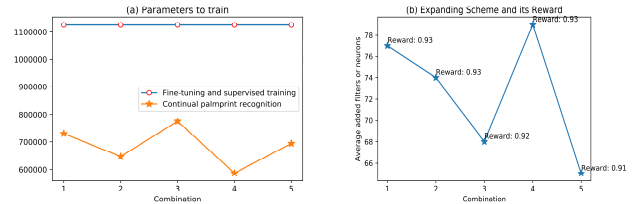


Fig.4 Model complexity and expanding scheme

5. CONCLUSIONS

In this paper, we propose a continual palmprint recognition system, which aims to help the biometric system adapt to new registered data quickly without catastrophic forgetting and high model complexity. We dynamically expand the neural network if necessary by an Actor-Critic-based reinforcement learning framework. Experiments conducted on incremental testing and cross domain palmprint recognition validate high accuracy and low training complexity of our model. Future works may be investigating new strategies for less training time and less space complexity.

6. REFERENCES

- [1] X. Qin, Y. Zhou, Z. He, Y. Wang, and Z. Tang, "A Faster R-CNN Based Method for Comic Characters Face Detection," in *2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*, 2017, vol. 01, pp. 1074-1080.
- [2] K. Cao and A. K. Jain, "Automated Latent Fingerprint Recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1-14, 2018.
- [3] X. Pan and Q. Q. Ruan, "Palmprint recognition using Gabor feature-based (2D) 2 PCA," *Neurocomputing*, vol. 71, no. 13, pp. 3032-3036, 2008.
- [4] Z. Guo, L. Zhang, D. Zhang, and X. Mou, "Hierarchical multiscale LBP for face and palmprint recognition," in *2010 IEEE International Conference on Image Processing*, 2010, pp. 4521-4524.
- [5] D. Zhong, Y. Yang, and X. Du, "Palmprint Recognition Using Siamese Network," in *Chinese Conference on Biometric Recognition*, 2018, pp. 48-55.
- [6] D. Zhong, X. Du, and K. Zhong, "Decade progress of palmprint recognition: A brief survey," *Neurocomputing*, 2018/08/20/ 2018.
- [7] L. Leng, F. Gao, Q. Chen, and C. Kim, "Palmprint recognition system on mobile devices with double-line-single-point assistance," *Personal and Ubiquitous Computing*, vol. 22, no. 1, pp. 93-104, / 2018.
- [8] L. Leng, M. Li, C. Kim, and X. Bi, "Dual-source discrimination power analysis for multi-instance contactless palmprint recognition," *Multimedia Tools Appl.*, vol. 76, no. 1, pp. 333-354, / 2017.
- [9] L. Leng and A. B. J. Teoh, "Alignment-free row-co-occurrence cancelable palmprint Fuzzy Vault," *Pattern Recognition*, vol. 48, no. 7, pp. 2290-2303, / 2015.
- [10] L. Leng, A. B. J. Teoh, M. Li, and M. K. Khan, "Orientation range of transposition for vertical correlation suppression of 2DPalmPhasor Code," *Multimedia Tools Appl.*, vol. 74, no. 24, pp. 11683-11701, / 2015.
- [11] J. Xu and Z. Zhu, "Reinforced Continual Learning," in *NeurIPS*, 2018, pp. 907-916.
- [12] B. Zoph and Q. V. Le, "Neural Architecture Search with Reinforcement Learning," *CoRR*, vol. abs/1611.01578, / 2016.
- [13] A. Kong, D. Zhang, and M. Kamel, "A survey of palmprint recognition," *Pattern Recognition*, vol. 42, no. 7, pp. 1408-1418, 2009.
- [14] A. Gumaee, R. Sammouda, A. Alsalman, and A. Alsanad, "An Improved Multispectral Palmprint Recognition System Using Autoencoder with Regularized Extreme Learning Machine," *Computational Intelligence and Neuroscience*, vol. 2018, pp. 1-13, 2018.
- [15] S. Minaee and Y. Wang, "Palmprint Recognition Using Deep Scattering Convolutional Network," *CoRR*, vol. abs/1603.09027, / 2016.
- [16] L. Dian and S. Dongmei, "Contactless palmprint recognition based on convolutional neural network," in *2016 IEEE 13th International Conference on Signal Processing (ICSP)*, 2016, pp. 1363-1367.
- [17] Z. Xie, Z. Guo, and C. Qian, "Palmprint gender classification by convolutional neural network," *IET Computer Vision*, vol. 12, no. 4, pp. 476-483, 2018.
- [18] J. Svoboda, J. Masci, and M. M. Bronstein, "Palmprint recognition via discriminative index learning," in *International Conference on Pattern Recognition*, 2017, pp. 4232-4237.
- [19] P. H. Pisani, A. C. Lorena, and A. C. P. L. F. d. Carvalho, "Adaptive Biometric Systems using Ensembles," *IEEE Intelligent Systems*, vol. 33, no. 2, pp. 19-28, 2018.
- [20] F. Shukur, "Using Multiagents for Context-Aware Adaptive Biometrics," in *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence (IJCAI)*, 2017, pp. 5209-5210.
- [21] A. S. Zacarias and L. A. Alexandre, "Overcoming Catastrophic Forgetting in Convolutional Neural Networks by Selective Network Augmentation," *CoRR*, vol. abs/1802.08250, / 2018.
- [22] F. Zenke, B. Poole, and S. Ganguli, "Continual Learning Through Synaptic Intelligence," in *ICML*, 2017, pp. 3987-3995.
- [23] C. Fernando *et al.*, "PathNet: Evolution Channels Gradient Descent in Super Neural Networks," *CoRR*, vol. abs/1701.08734, / 2017.
- [24] A. A. Rusu *et al.*, "Progressive Neural Networks," *CoRR*, vol. abs/1606.04671, / 2016.
- [25] J. Lee, J. Yoon, E. Yang, and S. J. Hwang, "Lifelong Learning with Dynamically Expandable Networks," *CoRR*, vol. abs/1708.01547, / 2017.
- [26] V. Mnih *et al.*, "Playing Atari with Deep Reinforcement Learning," *CoRR*, vol. abs/1312.5602, / 2013.
- [27] Z. Wang, T. Schaul, M. Hessel, H. v. Hasselt, M. Lanctot, and N. d. Freitas, "Dueling Network Architectures for Deep Reinforcement Learning," in *ICML*, 2016, pp. 1995-2003.
- [28] D. Zhang, Z. Guo, G. Lu, L. Zhang, and W. Zuo, "An Online System of Multispectral Palmprint Verification," *IEEE Transactions on Instrumentation & Measurement*, vol. 59, no. 2, pp. 480-490, 2010.
- [29] Z. Lin, L. Li, A. Yang, S. Ying, and Y. Meng, "Towards contactless palmprint recognition: A novel device, a new benchmark, and a collaborative representation based identification approach," *Pattern Recognition*, vol. 69, pp. 199-212, 2017.