Dynamic Hand Gesture Recognition using 2D Convolutional Neural Network

Yupeng Liu, Mingqiang Yang, Jie Li, Qinghe Zheng and Deqiang Wang

***Abstract*—In recent years, many algorithms arise in the field of dynamic gesture recognition. Traditional methods lack of accuracy and rely heavily on hand-crafted features. Due to powerful ability of feature extraction, deep learning methods show amazing performance, especially Convolutional Neural Network (CNN) that has been addressed lately in video analysis. However, some CNN-based methods such as C3D and Two-Stream are time-consuming and a great deal of computation is needed. In this paper we propose a novel method for hand gesture recognition based on 2D CNN. For a given video sequence, the input of the network is no longer a number of sampled images. We encode each sampled image firstly and encoding vector of each image is just got. For each feature vector encoded, we stack them to generate a new image that contains rich spatio-temporal information of gesture. The new image instead of origin video is then sent in traditional 2D CNN model and the classification result of gesture is finally obtained. 3D spatio-temporal information has been compressed into 2D presentation in the course of classification, and the computation and time to be consumed are reduced. At the same time, it reduces the risk of overfitting to a certain extent. Based on proposed method, the performance is evaluated on the Microsoft Research 3D dataset (MSR3D). The experiment shows that our approach is highly effective and efficient at classifying a wide variety of actions on MSR3D.**

***Index Terms*—dynamic gesture, action recognition, convolutional neural network**

I. Introduction

HAND gesture can be said to be another important communication tool besides human language, which contains abundant semantic information, is of great importance in computer vision and has a wide range of applications such as human computer interaction (HCI), augmented reality, affective computing, sign language recognition [1], [2], [3], [4], [5]. Gesture recognition mainly uses wearing equipment to directly detect the spatial position of hand and arm joints initially. Most of these devices connect the computer system with the user through wired technology, so that the user's gesture information can be transmitted to the recognition system without any error. The typical devices such as data gloves are also used [6], [7]. Owing to the dependence of equipment, its usability, comfort, safety and ease of use are poor. Therefore in recent years, more and more attention has been paid to the vision-based gesture recognition technology which can recognize gestures without touching devices [8], [9].

Gesture recognition technology can be divided into two kinds: static gesture recognition and dynamic gesture recognition. Static gesture recognition method can only recognize the state of gesture [10], [11], [12], [13], [14], but cannot perceive the continuous change of gesture. For example, if the hand is in a state of "fist", gesture can be correctly recognized. However the semantic actionłpalm slides from left to right, can be less distinguished. Dynamic gesture recognition method can process more complex gestures because it can acquire and fuse the temporal information of the movement [15], [16], so it has a broader application prospect in many domains.

In the last decade, there had been varied approaches without deep learning for gesture recognition, which mainly relied on hand-crafted features extracted on preprocessed frames [17] for representing spatio-temporal information like HOG [18], HOG3D [19], SIFT3D [20], HOF [21], IDT descriptors [22] and some temporal models like hidden Markov model (HMM) [23], conditional random fields [24], support vector machines (SVM) [25], Kalman filtering [26] and finite-state machine (FSM) [27], [28] were introduced. These conventional classifiers need features with strong expression ability, for which traditional methods are inadequate. Robustness classification of gestures under widely varying lighting conditions, and from different subjects or other complex scenes are still challenging problems now.

Apart from a collection of frames, a video can also be seen as a time series. Some of the most successful models for time series classification are recurrent neural networks (RNNs) with either standard cells or long short-term memory (LSTM) [29] cells. Their ability to learn dynamic temporal dependencies has allowed researchers to achieve breakthrough results in, for example, speech recognition [30], machine translation [31] , text classification [32] and image captioning [33]. Before feeding video to recurrent models, we need to incorporate some methods of spatial or spatio-temporal feature extraction. This point motivates the concept of combining CNNs with RNNs.

There are some differences between gesture recognition and general action recognition. For general video recognition datasets like UCF-101 [34] which consists of 13,320 videos of 101 human action categories, and Sports-1M [35] or HMDB-51 [36] etc, the temporal aspect is of less importance compared to spatial information of frames in these datasets. Gesture recognition pays more attention to the classification of actions. For example, the spatial appearance of a Piano

This paper was supported by National Key R&D Program of China (2018YFC0831503), The Fundamental Research Funds of Shandong University (2018JC040) and National Natural Science Foundation of China (61571275)

Yupeng Liu is with the School of Information Science and Engineering, Shandong University, Qingdao 266237, China (email: liuyupeng_work@163.com).

Mingqiang Yang is with the School of Information Science and Engineering, Shandong University, Jinan 250100, China (corresponding author, e-mail: imageinstitute@outlook.com).

Jie Li is with the School of Information Science and Engineering, Shandong University, Qingdao 266237, China (email: stu_jie_li@163.com).

Qinghe Zheng is with the School of Information Science and Engineering, Shandong University, Qingdao 266237, China (email:15005414319@163.com).

Deqiang Wang is with the School of Information Science and Engineering, Shandong University, Jinan 250100, China (e-mail: wdq_sdu@sdu.edu.cn).

almost certainly indicates the classification result is Playing Piano, since there is no other class involves a violin in UCF-101 dataset. The model has no need to capture motion information for this particular example. In the case of gesture recognition, however, motion knowledge plays a more critical role. Many gestures are not only defined by their spatial hand or arm placement, but also by their motion pattern.

Recently, deep convolutional neural networks have demonstrated their formidable power of extracting the discriminative features of images, and led to a series of breakthroughs for image classification [37], [38], [39], [40], [41], object detection [42], [43] and image retrieval [44] in image domain. But such deep features based on images are not directly suitable for videos due to lack of motion modeling. Compared to still images, the temporal component of video provides an additional clue for video-based tasks. So the challenge is to capture the complementary information on appearance from still frames and motion information between adjacent frames. Inspired by the potential of CNNs, two main algorithms: Two-Stream [45], [46] and C3D [47] have been proposed successively. Two-Stream method proposes a two-stream ConvNet architecture which incorporates spatial and temporal networks at once. The spatial stream, in the form of individual frame appearance, carries information about scenes and objects depicted in the video. The temporal part, in the form of motion across the frames, conveys the movement of the observer (the camera) and the objects. Despite its competitive performance, it is time-consuming and computational because of optical flow. C3D models use deep 3D ConvNet to learn spatio-temporal features. It encapsulates information related to objects, scenes and motion in a video simultaneously, and shows that C3D features outperforms the conventional 2D ConvNet features on various video analysis tasks.

In this work, we explore a variety of end-to-end trainable network for action classification applied to hand gesture recognition with 2D CNN. In this way network proposed can 1) crop the input both in spatial and temporal domain concurrently, which can be deemed to be a form of data augmentation 2) reduce the computation and time consuming of recognition, which also can decrease the risk of overfitting.

II. Related Works

Gesture recognition has been studied for decades and various algorithms for hand gesture recognition have emerged. We summarize some representative works in the following part.

A. Gesture Recognition without Deep Learning

Priyal *et al.* [11] separated hands from forearm regions, normalized the hand rotation using the geometry of gestures, and classified actions using a minimum distance, based on the Krawtchouk moment features which are found to be robust to viewpoint changes. Konečnỳ *et al.* [48] combined appearance features (Histograms of Oriented Gradients HOG) and motion descriptors (Histogram of Optical Flow HOF) with variants of a Dynamic Time Warping (DTW) technique for parallel temporal segmentation and recognition. Wu *et al.* [49] performed morphological denoising on depth images and automatically segmented the temporal boundaries, used Maximum Correlations Coefficient approach based on features extracted by Extended Motion History Image (Extended-MHI) with Multi-view Spectral Embedding (MSE) algorithm which was used to fuse duo modalities in a physically meaningful manner to classify gestures. Lui *et al.* [50] characterized action videos as data tensors and demonstrated their association with a product manifold and used the least squares regression algorithm for hand gestures recognition. Wan *et al.* [51] proposed spatio-temporal features named 3D SIFT, 3D Sparse Motion SIFT (3D SMoSIFT) scale-invariant feature and 3D enhanced motion SIFT (3D EMoSIFT) in detail, from RGB-D data and employed a bag of visual words model (BoVW) for activity recognition. Wang *et al.* [52] used Hidden Markov Models (HMMs) algorithm for dynamic gesture trajectory modeling and treated invariant curve moments as global features and orientation as local features to represent the trajectory of hand gesture for recognition. Wan *et al.* [53] proposed a novel spatio-temporal feature which are shown to be invariant and robust to scale, rotation and partial occlusions, namely mixed features around sparse key points (MFSK) for one-shot learning gesture recognition. Oreifej *et al.* [54] used a histogram to describe the depth sequence, which is designed to capture the distribution of the surface normal orientation in the 4D space of time, depth, and spatial coordinates.

B. Action Recognition with Deep Learning

With supervised training, neural networks have demonstrated remarkable performances on video classification dataset. Karpathy *et al.* [35] proposed a CNN-based model with multi-resolution to classify videos on large-scale datasets, showing significant performance improvements compared to strong feature-based baselines. Ji *et al.* [55] extracted features from both the spatial and the temporal dimensions by performing 3D convolutions, thereby capturing the motion information encoded in multiple adjacent frames. Simonyan *et al.* [45] proposed a two-stream ConvNet architecture which incorporates spatial network trained on origin single frame and temporal network trained on multi-frame dense optical flow, and used multi-task learning to increase the amount of training data. Tran *et al.* [47] proposed a simple, effective approach for spatio-temporal feature learning using deep 3-dimensional convolutional networks (3D ConvNets) trained on a large scale supervised video dataset, and achieved competitive performance. Based on Two-Stream, Wang *et al.* [56] used a novel framework named temporal segment network (TSN), which aimed to capture long-range temporal structure, with multi input modalities for video classification. Molchanov *et al.* [57] used a CNN-based classifier consisting of two sub-networks: a high-resolution network (HRN) and a low-resolution network (LRN). The two networks were fused by multiplying their respective class-membership probabilities element-wise. Pigou *et al.* [58] explored CNN based architectures for gesture recognition and proposed a new end-to-end trainable neural network method incorporating temporal convolutions and bidirectional recurrence. Li *et al.* [59] trained a CNN architecture based on a soft attention mechanism in an end-to-end manner, which was capable of automatically localizing hands and classifying hand gestures.

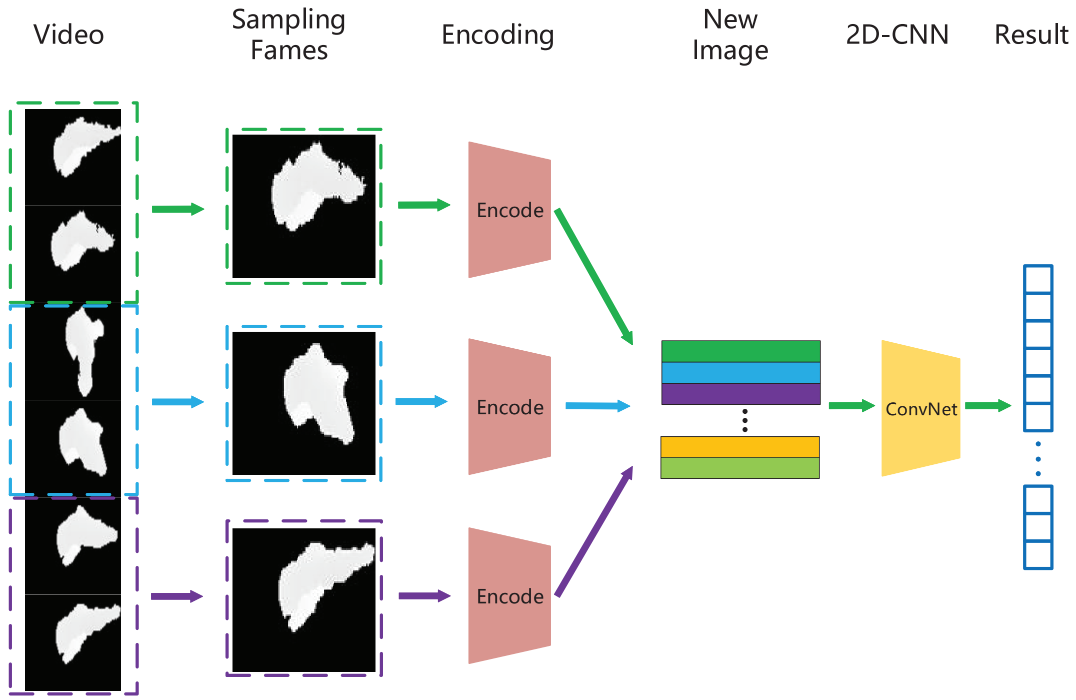

Fig. 1. **The framework of our approach.** We use a 2D CNN-based model for gesture recognition. The input is an image sequence representing the whole action. The classifier is composed of two mainly module: encoding module and 2D CNN. The encoding module encodes each image to a feature vector, which is fused with other vectors in the following and becomes a part of the new image. The newly generated image X with spatio-temporal information is then sent to 2D CNN for further processing. Moreover, the encoding methods can vary a lot.

III. Approach

In this section, we will introduce the construction of our method in detail and explain the rationality of the structure from many aspects. We first introduce the overall framework of our approach. Then we elaborate some parts of our architecture for training. Finally, we explore good practices for data augmentation in order to make network perform better.

A. Architecture

In Temporal Segment Network (TSN), the input is no longer single frame or frame stack, but a sequence of short snippets sparsely sampled from the entire video. By this way, it can enable efficient and effective learning by using the whole video to model long-range temporal structure. Motivated by TSN, we propose a novel framework for hand gesture recognition as shown in Fig. 1 based on the sparse temporal sampling strategy.

A given video V is divided into K segments $\{S_1, S_2, \ldots, S_k\}$ with uniform length (K is hyper parameter). Then we get image sequence $\{T_1, T_2, \ldots, T_k\}$ from K segments, $T \in \mathbb{R}^{h\times w\times c}$. T_k is randomly sampled from S_k which means that T_K and S_k are one-to-one correspondence. As shown in Fig. 1, encoding module is responsible for encoding image T_k to the vector V_k which represents feature of T_k, $V \in \mathbb{R}^{h'\times w'\times c'}$. We denote encoding process as $\mathcal{F}(T_k)$ function which operates on the sampled image T_k.

$$V_k = \mathcal{F}(T_k), k = 1, 2, \ldots, K \tag{1}$$

Then the new image called X (containing temporal information) is composed of vectors obtained before. Each row of image X is the vector encoded from T_k.

$$X = V_1 \diamond V_2 \diamond \ldots \diamond V_k, k = 1, 2, \ldots, K \tag{2}$$

where $\diamond$ means concatenation.

B. Input and Encoding Module

From Fig. 2, we can see that the image X contains spatio-temporal information of hand gesture action. The row direction of the image X preserves spatial information and the column direction encodes temporal information. Finally, image X is sent to 2D CNN and gesture classification result is got finally.

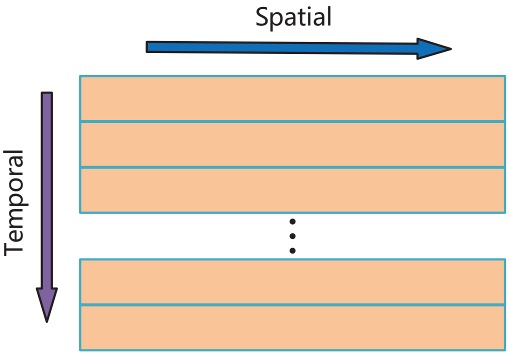

Fig. 2. **New image**. As we mentioned before, encoding methods can vary a lot. For convenience, we just resize each image to a row vector with a fixed size. Then each vector is stacked with others in order to form an image. The row direction of the new image preserves spatial information and the column direction encodes temporal information.

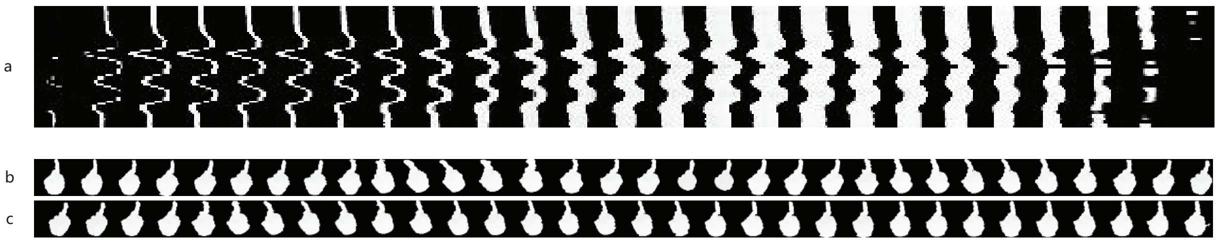

Fig. 3. **Visualization of encoding**. a), b) and c) are from the same image sequences. a) means the new image, of which the elements are from b) and c). b) means frames with a sequence number no more than 32. c) means frames with a sequence number more than 32 but less than 65.

TABLE I
Architecture of CNN.
The net has three convolution, three max-pooling, and two fully connected layers, followed by a softmax output layer. Due to abnormal size of input, we should be careful to adopt the size of kernel in each layer.

Input	64×625
Conv1	Channel: 64; Kernel: 3×7
Pooling	Type: Max; Kernel: 2×2
Conv2	Channel: 64; Kernel: 3×5
Pooling	Type: Max; Kernel: 2×2
Conv3	Channel: 128; Kernel: 3×3
Pooling	Type: Max; Kernel: 2×2
FC1	512
FC2	12

Our proposed network structure is depicted in Table. I. There are a total of three convolution layers and two fully connection layers. In general, the number of columns in the input is larger than that of rows. So in the first convolution layer, 64 kernels of size 3×7 with a stride of 1 pixel are utilized to capture enough spatial information with a large receptive field in the row direction. The second convolution layer takes the output of the first convolution layer as input with filters of size $64 \times 3 \times 5$. The third convolution layer has 128 kernels of size 3×3. The first fully-connected layer has 512 neurons to obtain enough semantic information. And the last fully-connected layer produces a distribution over 12 class labels. The final classification result is got with maximum output value. What the table fails to depict is that Rectified Linear Unit (ReLU) layer follows each convolution layer, which has shown to bring about acceleration on training. Due to the number of data, overfitting tends to occur in neural network's training process. To tackle with overfitting problem, we also introduce Batch Normalization layer and Dropout layer into our network, which are shown to be greatly effective and efficient.

The input of 2D-CNN is new image X containing spatio-temporal information, which is shown in Fig. 3. As mentioned in part A, the spatial information is embedded in each row of X and the temporal information is recorded in column direction. As we can see in Fig. 1, the input is computed on the output of encoding module which produces a row vector. From the front of the X, we note that the finger swings around many times. And based on the rest of the image X, we can observe that the palm is barely moving. So the hand gesture action can be inferred from new image X, which corroborates the validity of our idea in this paper.

Considering the convenience and computation, for each frame T_k in sampled image sequence, we just resize it with a fixed size like 30×30. Then resized image is flattened for stacking. Though this is a simplified implementation of the encoding module, it can still achieve competitive performance on the MSR Gesture 3D dataset [60]. There are also many encoding methods can be tried, such as ConvNet and traditional feature extraction algorithms like FHOG. Due to the amount of data, mentioned methods above are very prone to overfitting. And with a large number of parameters, this phenomenon becomes more frequent to occur.

C. Regularization Techniques

1) Batch Normalization: The fact that the input distribution of each layer in the training process of the deep neural network changes with the parameters of the previous layer makes training become hard. It is prone to bring about gradient explosion or gradient vanishing problem, which seem to be the main obstacles to train deep neural networks. And it also makes training more difficult that saturating nonlinearities often occurs. We call this phenomenon as the internal covariate shift. Before Batch Normalization [61], [62] is brought forward, it is common and necessary for us to adopt lower learning rate, take care of parameter initialization and attempt more activation functions like ReLU, PReLU. Batch Normalization can address this problem to a great extent. It enables us to apply much higher learning rate and be less careful about weights initialization in training process. Resembling dropout technology, BN puts some noise on each hidden layers activation value to introduce more randomness into network. It has been shown that BN is effective to alleviate overfitting problem because it can bring about a slight regularization effect for networks.

In our framework, we use Batch Normalization layer and Dropout layer for tackling with overfitting problem and employ higher learning rates to accelerate network training.

2) Dropout: Overfitting is a noteworthy and severe problem in deep neural networks. As Srivastava [63] referred, dropout is a technique for dealing with this troublesome problem. It is achieved by randomly dropping some neural units (along with their connections) with an artificial designed ratio in the neural network during training. The degree of co-adapting between neural units just are abated to a great extent. Applying dropout to a neural network amounts to extracting a thinned network from the origin whole network. In training process, a series of thinned networks are sampled by using dropout, which is under a certain dropout ratio. In test phase, it is not feasible to

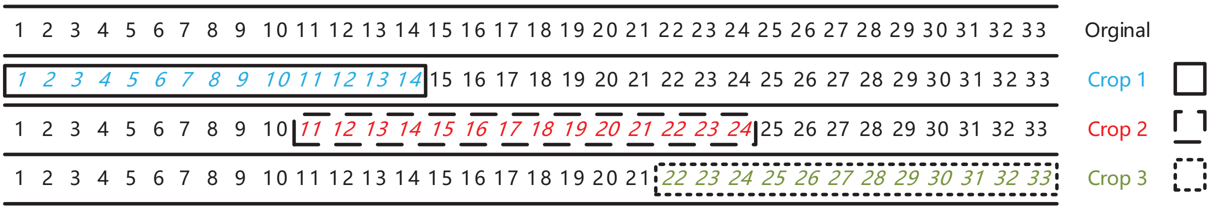

Fig. 4. **Examples of temporal data augmentation**. The numbers in each row are denotes the frame IDs in the original sequences. Three patches select three different segments of the frame sequence, which carry unique temporal information.

directly calculate result by averaging the predictions from exponential thinned models explicitly. We should use a whole unthinned network which just has smaller weights to predict results by averaging the predictions of all those thinned networks implicitly. Dropout significantly reduces overfitting and performs outstandingly compared to other regularization approaches. The neural network model with dropout will be expounded in the following.

From a certain point of view, dropout can be deemed to train a great deal of neural networks with shared parameters and employ bagging concept which is popular in ensemble learning theory in test phase for the sake of better generalization. From another perspective, it seems to make sense that dropout is explicated as a kind of data augmentation operation in the space of few input without prior knowledge.

Due to limited video data, we add an extra dropout layer after $fc1$ layer in our architecture to further reduce the effect of overfitting. The dropout ratio is set as 0.8 after a multitude of trials.

D. Data Augmentation

1) Crop: Apart from BN and dropout regularization techniques, the easiest and most commonly used way to reduce overfitting is to artificially enlarge the dataset with label-preserving transforms. It is common to do augmentation online while training. Cropping operation has shown to be an effective way to augment data while lifting network performance.

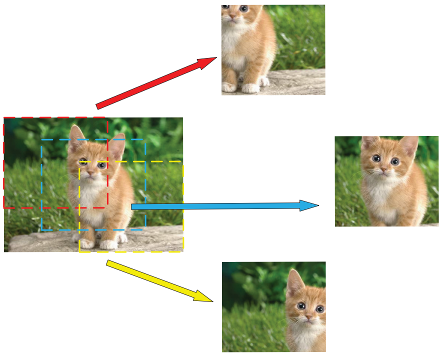

Fig. 5. **Traditional cropping** is carried on spatial domain. Though missing part of information, the key part that helps the task will be preserved finally. Three patches by cropping original image are somewhat different from one another. We still can recognize the object in the three patches as cats.

In traditional 2D CNN, we do crop by extracting patches with fixed size from origin image and using these extracted patches as network inputs for training in Fig. 5. Though lack of some information about object due to cropping operation, we also note that crux of the image for classification or detection is still preserved. It is still capable of recognizing the object based on remaining patches. Randomly cropping in training process is usually used as a regularization method. But in test phase, it is common to crop image based on the center.

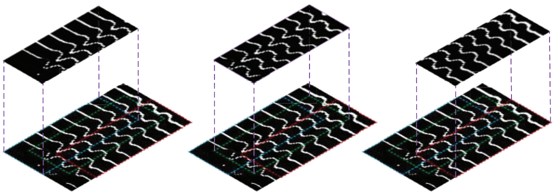

Fig. 6. **Visualization of cropping in spatial and temporal domain simultaneously**. As mentioned before, the row direction and the column direction of the new image preserves spatial and temporal information respectively. The crop in the column direction amounts to selecting part of consecutive frames in video. The crop in the row direction is same as crop one region of each frame.

In our method, cropping new image X means we can crop video sequence in spatial and temporal domain simultaneously. From Fig. 6, three different cropped patches have their own unique spatio-temporal information with a little difference with the others. Firstly, in spatial domain, patch #1 extracts the upper part of the image and obtains the appearance of the fingertip. Patch #2 mainly consists of the middle part of the X and fingers exterior is still preserved. And patch #3 is drawn from the lower part of finger, which may have almost same amount of information compared to other patches. In temporal domain, different patch occupies different time ranges but same length of time in Fig. 4. For whole action video sequence, patch #1 describes hand gesture action from frame 1 to frame 14, namely the first half of the action. We can see hand waves around in patch #2 without being standstill, in other words patch #2 includes useful motion information without inefficacious frame. Similarly, patch #3 involves movement of hand from frame 22 to frame 33. It is important that all of patches are in a position to represent the motion of hand in the absence of little information.

2) Image Transformation: For the sake of further preventing overfitting and increasing the generalization performance of the network, we also do some spatial transformations on image X and frames that are sent into encoding modules. Methods of transformation varies a lot. Due to characteristics of gesture recognition problems, many of them cannot be used. If some of images in video sequence are flipped horizontally, the quondam action may be identified as another one, such as move to right and move to left actions. Analogously, translation operation is also prone to same problem.

In our paper, we apply two transformation ways to augment training data. Firstly, rotation operation is applied on image X after encoding module. To avoid making huge changes to the image so as to affect the classification, we limit the degree of rotation in practice: $\pm 10^\circ$. Secondly, Gaussian noise is also utilized to enlarge dataset. When neural network tries to learn high-frequency features that may not be useful, overfitting often occurs. Gaussian noise with zero-mean essentially has data points on all frequencies, which can effectively distort the high-frequency feature and weaken its impact on the model. This also means that the low-frequency components (usually the data we care about) can be distorted, but neural networks can learn to ignore this part of the impact. Adding the appropriate amount of noise can enhance the learning ability and generalization of the neural network.

IV. Experiments

In this section, we first give a description of MSR Gesture 3D dataset, on which our experiments are evaluated. Secondly, we carry on elaboration regarding implementation of our approach. Finally, we show the experimental results and compare the performance of our method with other methods.

A. Dataset

The MSR Gesture 3D dataset contains 336 files in total, each corresponding to a depth sequence, and is considered challenging because of self-occlusions. And the hand portion (above the wrist) has been segmented. There are 12 dynamic hand gestures defined by the American Sign Language (ASL). It contains 10 subjects totally, each of them performing each gesture 2 or 3 times. This dataset presents more self-occlusions than other datasets. Some examples in the dataset are shown in Fig. 7. Notice that although this dataset contains both the color and depth frames, only depth frames are used in our experiments. We follow experiment protocol proposed: using recordings of 9 persons for training and the remaining person for testing that is also to say leave-one-out cross-validation strategy. We also note that the resolution of depth map is different from one sequence to another. In order to ensure the consistency of the scale, each depth frame in all sequences is resized to the same size.

Fig. 7. **Action examples on the MSR Gesture 3D dataset**. We sample a number of image sequences for demonstration. a), b) and c) means “Pig”, “Where” and “Z” gesture respectively.

B. Implementation Details

Our implementation for 2D ConvNet is derived from the publicly available Caffe toolbox, and network is trained on Geforce 1050TI GPU, Intel(R) i5-7500 CPU @ 3.40GHz. In this part, we give a description of the implementation details of our scheme.

1) ConvNet configuration: The configuration of layers in network is schematically shown in Table. I. Because there are only three convolution layers and two fully-connected layers, we cannot employ pre-trained CNN model trained on ImageNet dataset. Therefore, we just train the network from scratch. In consideration of the size of image X in which the number of columns is larger than that of rows in practice, it is essential for us to be careful when we design the size of kernel. As similar with other works, all convolution layers is followed by the rectification (ReLU) activation function; each max pooling layer is performed over kernel of size 33 with stride 2; the only two fully-connected layers also use ReLU function; batch normalization layers employ the same settings as recommended in Caffe manual; the dropout ratio is set to 0.8 in dropout layer. The size of all kernels in convolution layers and max pooling layers are shown in Table. I. Different with previous works, which generally initialize the weights in each layer from a zero-mean Gaussian distribution with standard deviation 0.01, we employ Xavier initialization scheme which is proved to be efficient and brings substantially faster convergence.

2) Training: We use the mini-batch stochastic gradient descent (SGD) algorithm to train our model and learn the network parameters, where the batch size is set to 64, momentum set to 0.9, and weight decay of 0.0005. Weight decay is important for network training. It is not merely as a regularization technique for cost function, which can avoid overfitting, but also reduces the networks training error. The update formula of weight is

For learning rate, all layers are the same, which are adjusted by step in SGD during training. It is natural in previous works to divide the learning rate by 10 or 5 when the network accuracy in validation dataset no longer increases with current learning rate. In this paper, we no longer employ traditional approach but multi-step strategy, which is similar to step but allows non uniform steps defined by stepvalues. And the learning rate is initialized as 0.001, which reduces to its $\frac{1}{10}$ after each of stepvalues iteration.

For the input, we only employ RGB frames without consideration of the stack of optical flow frames, which greatly lessens the amount of computation and time-consuming. According to the length of video in dataset, we resample the video sequence to the same length $L = 128$. With regard to video with a length of less than L, we just copy some frames in this video repeatedly to satisfy the need. For video whose length is more than L, we randomly sample frames from video to fulfil the purpose. As mentioned in Section 3, we randomly sample $K = 64$ frames from L frames. Regarding to K frames, each of them is resized as 25×25. Then K resized frames compose image X, which is just the input of 2D CNN.

3) Testing: In the testing phase, for a given video, the same manipulation as the training phase is performed except for crop operation. Randomly cropping strategy is replaced by center cropping. Namely, we just crop the center of the image. As mentioned in Section *A*, we follow the official test criterion: leave-one-out cross-validation strategy. For all people in dataset, we use one people for validation and others for training.

C. Results

After giving a description of dataset and implementation details, we compare our approach with a number of methods over MSR 3D dataset. We report the average accuracy only instead of cross-validation results. Our results on MSR Gesture 3D are summarized in Table. II with comparison against other traditional methods.

TABLE II
GESTURE RECOGNITION RESULTS.

Method	Accuracy(%)
SR (Sparse Representations)	83.63
GSR [64]	85.42
UMLD [60]	85.2
APADS+HMM [65]	80.7
ROP [66] (Without sparse coding)	86.5
ROP [66] (Sparse coding)	88.5
SVM on Raw Features	62.77
HON4D [54]	87.29
HON4D + D_{disc} [54]	92.45
AG [60] (Occupancy Features)	80.5
AG [60] (Silhouette Features)	87.7
3D CNN [55]	69
Ours	93.81

Our proposed approach achieves top-1 accuracy of 93.81%. As can be seen from Table. II, the proposed method performs significantly better than the traditional methods based on carefully designed features except for HOG4D. The model with lowest accuracy is SVM on raw features, of which the features are not representative and compact. And the performance gap of SVM on raw features is small 31.34% compared to our method. It indicated that the feature for gesture recognition is a major factor. And we use SVM method as baseline in this paragraph. APADS+HMM used Hidden Markov Models (HMMs) for dynamic gesture trajectory modeling and recognition, treating invariant curve moments as global features and orientation as local features to represent the trajectory of hand gesture. With all these features and trajectory model, APADS+HMM achieved a gain of 18.94%. ROP method extracted semi-local features called random occupancy pattern (ROP) features to enhance the expression ability of feature, which was shown to have a positive impact on recognition with 24.34% gain in performance. Due to a sparse coding approach, ROP (with sparse coding) has a better result, however its accuracy rate is still lower than 90%. The algorithm with highest accuracy rate in methods without deep neural network is based on HON4D features. It used a histogram to capture the distribution of the surface normal orientation in the 4D space of time, depth, and spatial coordinates. And the result proved the effectiveness and superiority of HON4D feature with 92.45% accuracy rate. In spite of such a great improvement, it is still not as high as our method.

TABLE III
LEAVE-ONE-OUT CROSS-VALIDATION RESULT.
OUR PROPOSED APPROACH IS CARRIED ON TEN DIFFERENT SUBJECTS.

Subject Number	Accuracy(%)
Subject 1	83.33
Subject 2	83.33
Subject 3	100
Subject 4	100
Subject 5	91.67
Subject 6	100
Subject 7	97.22
Subject 8	90.91
Subject 9	91.67
Subject 10	100

We compare traditional methods in the last paragraph, and in this part we are going to give a comparison between 3D CNN and our method. Shui Ji *et al* [55]. developed a novel 3D CNN model for action recognition to address the problem that CNNs are currently limited to handle 2D inputs. 3D CNN model extracts spatio-temporal features from both spatial and temporal dimensions, thereby capturing the motion information encoded in multiple adjacent frames. As we can see in Table. II, the result obtained by 3D CNN is not ideal. We use 2D CNN with less parameters and achieve top accuracy rate. Moreover, our method clearly shows the power of 2D CNN in gesture recognition even video classification domain.

From Table. III, we can see the cross-validation results on ten subjects. On four subjects (3,4,6,10), best results are got by our model with 100% accuracy. The results with lowest accuracy are performed on subject 1 and subject 2, but they are still higher than eighty percent. The accuracies on the other subjects range from approximately ninety percent to about ninety-seven percent. Our method achieves 93.81% recognition accuracy on this challenging dataset finally after averaging accuracies.

D. Learning Curves

It is common for us to analyse the performance of model in training process via loss or accuracy curve. And loss curves can be conductive to make adjustments to the model for raising recognition accuracy. In practice, we draw loss on mini-batch data after each iteration. Loss on whole train set and whole test set are drawn with preseted interval that is set to 50. As shown in Fig. 8, we can observe that the network does not overfit even after a large amount of iterations. In the beginning, all the losses are a little high and we can see the losses rapidly fall off along with the increasement of iteration. At this stage, the descend trend of each loss basically keeps pace with one another, which lasts until about 2000 iterations. Then the rate of loss reduction slows down till 4000 iterations approximately. In this phase, train loss is lowest among the three losses, which is almost close to 0, and the model seems to have fitted train set well. Furthermore, mini-batch loss is approximately equal to test loss ignoring dramatical fluctuations of mini-batch loss. After 4000 iterations, all losses tends to be steady without obvious undulation, which shows us that the training has been done.

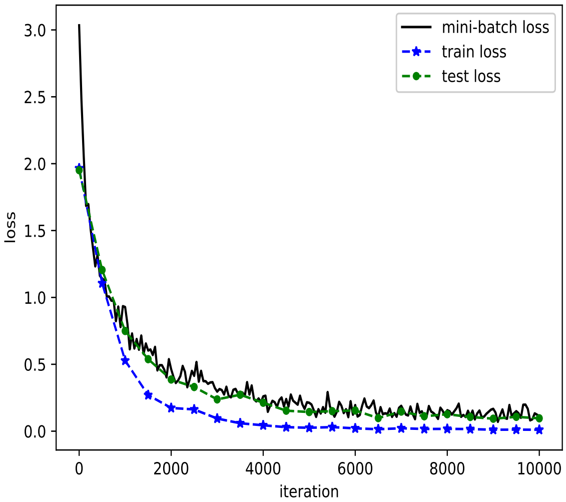

Fig. 8. **Loss curves**. Mini-batch loss means the loss on the mini-batch data. Train loss and test loss represents the loss on the whole train dataset and the loss on the whole test dataset, respectively.

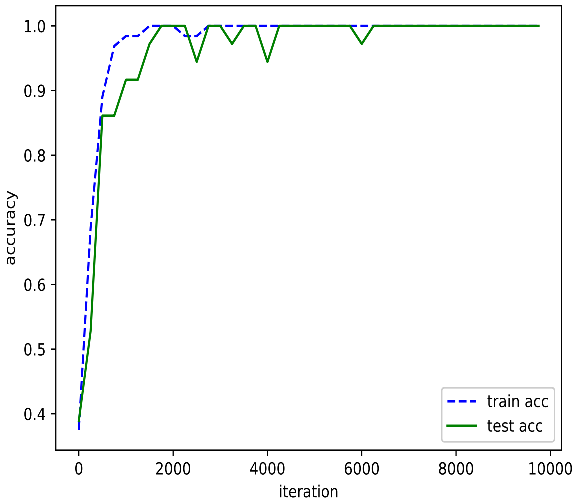

Fig. 9. **Accuracy curves**. Train accuracy and test accuracy represents the accuracy on the whole train dataset and the accuracy on the whole test dataset, respectively.

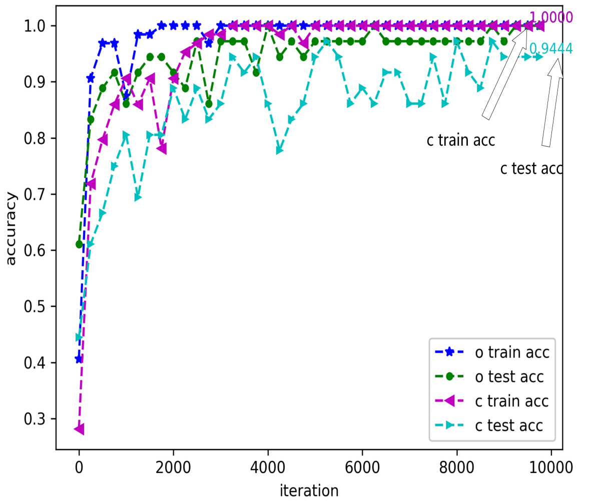

Fig. 10. **Accuracy curves**. 'o' denotes original model without mirror operation, and 'c' denotes changed model in which the input of network is mirrored.

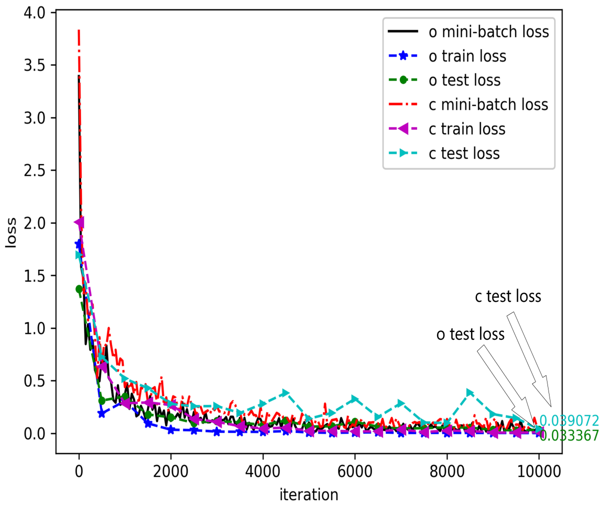

Fig. 11. **Loss curves**. 'o' denotes original model without mirror operation, and 'c' denotes changed model in which the input of network is mirrored.

Fig. 9 shows how an increase of accuracy in learning. Before about a thousand of iterations, test accuracy has been rapidly increasing while the train loss and test loss have been falling. Similar to the loss curve, test accuracy keeps pace with train accuracy by and large. Then the difference of accuracy between train set and test set commences. Performance on test set is a little bit worse than that of train set and test accuracy shows strong volatility. Starting from the 2000 iterations, train accuracy rate tends to be steady while the downward trend of the loss is still maintained. From approximately four thousand iterations, all the accuracy curves perform steadily with only minor fluctuations, which shows similar trend compared to loss curves.

1) Mirror: In the general action recognition task, the contribution of temporal information to the target task is small. Most of samples can be accurately identified only by recognizing the object of the action, such as "playing the violin". As long as the violin can be recognized in the image sequence, the algorithm can be more likely to classify correctly. For gesture recognition task, the importance of temporal information is major. Gestures in the same shape and with different movement directions can represent different classes of action. In Fig. 10, 'o' (original) denotes original model without mirror manipulation, 'c' (changed) denotes changed model which randomly mirrors the input of the network. The accuracy of the original model is between 97% and 100%, while that of the modified model is stable at 94% with large amplitude of curve jitter. In Fig. 11, the loss curve of the testset of the 'o' model tends to be stable after the 4k-th iteration, and loss value of testset approaches that of the training set. For 'c' model, the loss curve of the test set fluctuates greatly, and the loss value of the peak is larger,

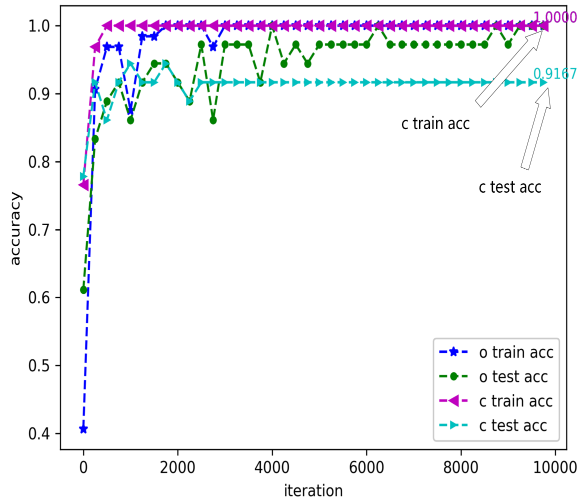

Fig. 12. **Accuracy curves**. ‘o’ denotes original model in which input is cropped in spatio-temporal domain, and ‘c’ denotes changed model in which the input is only cropped in spatial space.

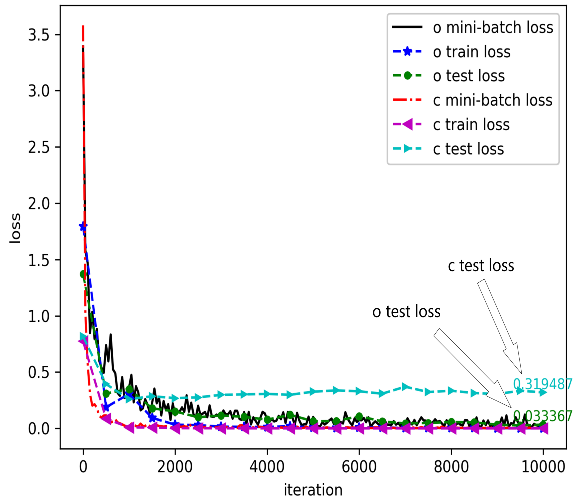

Fig. 13. **Loss curves**. ‘o’ denotes original model in which input is cropped in spatio-temporal domain, and ‘c’ denotes changed model in which the input is only cropped in spatial space.

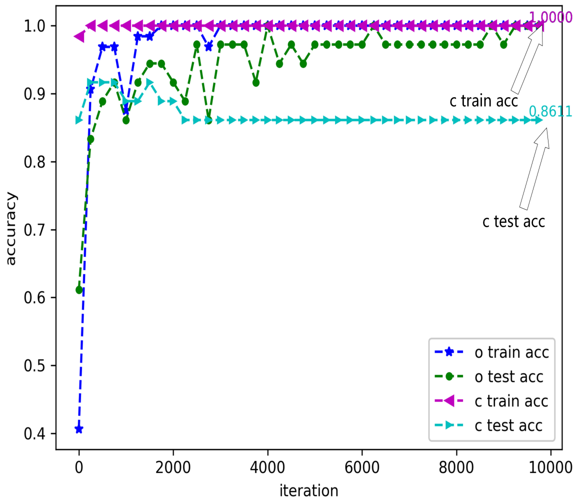

Fig. 14. **Accuracy curves**. ‘o’ denotes original model in which input is cropped in spatio-temporal domain, and ‘c’ denotes changed model without any cropping operation.

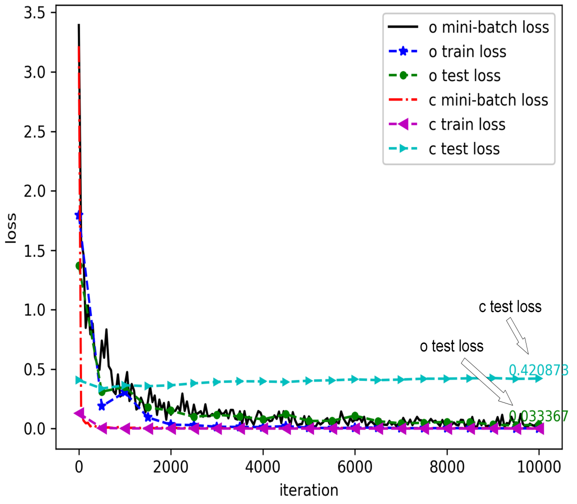

Fig. 15. **Loss curves**. ‘o’ denotes original model in which input is cropped in spatio-temporal domain, and ‘c’ denotes changed model without any cropping operation.

which shows us that the effect of model training is not up to expect. As mentioned above, for gesture recognition task, image mirror operation changes the sample to a large extent, which is not conducive to the training of the model.

2) Crop in temporal domain: Spatio-temporal cropping is propitious to the training of model. In this experiment, we only crop input in spatial domain in training to verify the effectiveness of temporal domain cropping. As shown in Fig. 12, ‘o’ denotes the original model, and samples are cropped in spatio-temporal domain. ‘c’ denotes the modified model, in which samples are only cropped in space domain, but not in time domain. Although the accuracy of the test set of the ‘c’ model is 91%, there is still a gap of nearly 10% between the accuracy of the test set and that of the training set. As shown in Fig. 13, the loss value of the ‘c’ model on the test set kept at about 0.3, while the loss on the training set is close to 0. It can be seen that there is a slight over-fitting problem in the training of ‘c’ model. Therefore, cropping in temporal-domain of video data is conducive to the training of the model, and can improve the performance of the model.

3) Crop in spatio-temporal domain: The cropping operation of input image increases the randomness of samples, which can be seen as a way of data augmentation, and can reduce the risk of over-fitting in the process of model training to a large extent. In this experiment, we compare the classification results of spatio-temporal cropped and unchanged samples. As shown in Fig. 14, ‘o’ is the original model, and the sample is cropped in the spatio-temporal domain; ‘c’ is the modified model, without any cropping operation of the sample. For the ‘c’ model, after the 2k-th iteration, the accuracy of the test set is as low as 86%, while the accuracy of the training set is as high as 100%, which has obvious over-fitting phenomenon. As shown in Fig. 15, the loss values of training set and test set of the ‘o’ model remain approximate throughout the training process, which indicates that the training of the model has reached a good state. Compared with the ‘o’ model, the loss value of the ‘c’ model in the training set decreases faster at the beginning, while the loss value of the test set is very high. In the whole training process, there is a big gap between the performance of ‘c’ model in test set and training set, which shows that over-fitting problem emerges. Therefore, cropping in spatio-temporal domain is extremely beneficial to the training of the model, and improve the performance of the model greatly.

E. Confusion Matrix

Confusion matrix is a specific table layout that supports visualization of the classification results of an algorithm mostly used in supervised learning. Each row of the confusion matrix represents the instances in a predicted class while each column represents the instances in an actual class. Element $M_{(i,j)}$ in confusion matrix denotes the number of samples in class i that are assigned to class j. We can analysis the relevance between arbitrary two categories by confusion matrix.

The confusion matrix is shown in Fig. 16. The value of the matrix has been normalized in the range of 0 to 1 for convenience. As we can see, the proposed method performs quite well for most of the gestures according to confusion matrix because the colors of the clino diagonal are closest to red. It can be observed that the most confused pair among categories is store gesture and milk gesture. We finally draw a conclusion from the observation of training data shown in Fig. 17. The sign for “store” is made by bending both wrists and pointing both hands down and it is common to pivot both of hands forward (away from body) twice. The sign for “milk” is made by forming a “C” and closing it twice into an “S” hand. In subject #1, the video representing “milk” action only captures one hand of the gesture, which makes the “milk” gesture very similar to the “store” gesture relative to other gestures due to the resemble movement of the palm.

V. Conclusion

In this paper, we present a novel 2D CNN-based framework that aims to capture spatio-temporal feature with less computation and time-consuming for hand gesture recognition. By encoding sampled frames from a video sequence to a new image which is sent to the 2D CNN, we use less parameters and reduce the computation cost of the network, easily employing temporal jittering and augmenting data by cropping the new image. As demonstrated on MSR Gesture 3D dataset, this work has brought competitive performance while maintaining a reasonable computational cost.

Our work aims to verify the feasibility of using 2D CNN for gesture recognition, instead of depended on optical flow. At the same time, we can use other techniques such as key frame extraction, adding spatial subnetwork to enhance the performance because our network mainly concentrates on the motion, which may result in the lack of spatial information.

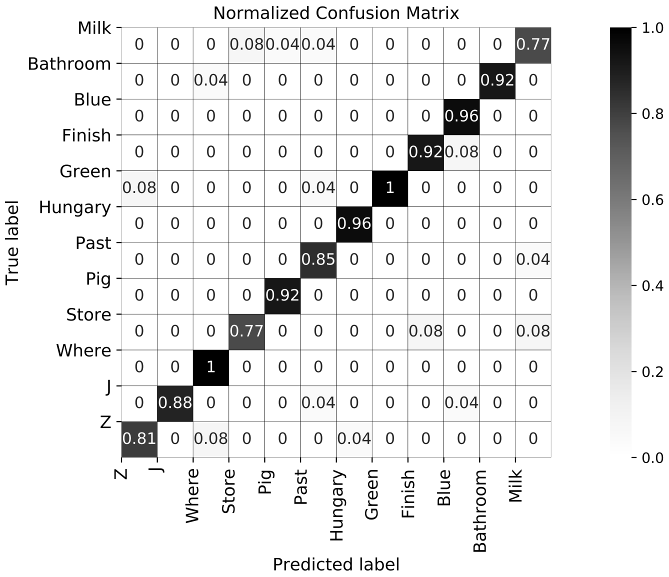

Fig. 16. **The confusion matrix for our proposed gesture classifier**. The size of the value is expressed by color.

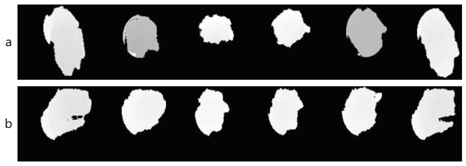

Fig. 17. **The sample frames of the confused gestures**. a) the “Store” gesture. b) the “Milk” gesture. We can see the “Store” action in which the palm bobs up and down is similar to the “Milk” gesture in which the palm is closed first and then released.

References

[1] S. Gupta, P. Molchanov, X. Yang, K. Kim, S. Tyree, and J. Kautz, “Towards selecting robust hand gestures for automotive interfaces,” in *2016 IEEE Intelligent Vehicles Symposium (IV)*, 2016, pp. 1350–1357.

[2] B. Feng, F. He, X. Wang, Y. Wu, H. Wang, S. Yi, and W. Liu, “Depth-projection-map-based bag of contour fragments for robust hand gesture recognition,” *IEEE Transactions on Human-Machine Systems*, vol. 47, no. 4, pp. 511–523, 2017.

[3] P. Molchanov, X. Yang, S. Gupta, K. Kim, S. Tyree, and J. Kautz, “Online detection and classification of dynamic hand gestures with recurrent 3d convolutional neural network,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 4207–4215.

[4] Z. Ren, J. Yuan, and Z. Zhang, “Robust hand gesture recognition based on finger-earth mover’s distance with a commodity depth camera,” in *Proceedings of the 19th ACM international conference on Multimedia*, 2011, pp. 1093–1096.

[5] T. Yamashita and T. Watasue, “Hand posture recognition based on bottom-up structured deep convolutional neural network with curriculum learning,” in *2014 IEEE International Conference on Image Processing (ICIP)*, 2014, pp. 853–857.

[6] J.-H. Kim, N. D. Thang, and T.-S. Kim, “3-d hand motion tracking and gesture recognition using a data glove,” in *2009 IEEE International Symposium on Industrial Electronics*, 2009, pp. 1013–1018.

[7] O. Luzanin and M. Plancak, “Hand gesture recognition using low-budget data glove and cluster-trained probabilistic neural network,” *Assembly Automation*, vol. 34, no. 1, pp. 94–105, 2014.

[8] S. S. Rautaray and A. Agrawal, “Vision based hand gesture recognition for human computer interaction: a survey,” *Artificial Intelligence Review*, vol. 43, no. 1, pp. 1–54, 2015.

[9] P. K. Pisharady and M. Saerbeck, “Recent methods and databases in vision-based hand gesture recognition: A review,” *Computer Vision and Image Understanding*, vol. 141, pp. 152–165, 2015.

[10] H. Zhou, D. J. Lin, and T. S. Huang, “Static hand gesture recognition based on local orientation histogram feature distribution model,” in *2004 Conference on Computer Vision and Pattern Recognition Workshop*, 2004, pp. 161–161.

[11] S. P. Priyal and P. K. Bora, "A robust static hand gesture recognition system using geometry based normalizations and krawtchouk moments," *Pattern Recognition*, vol. 46, no. 8, pp. 2202–2219, 2013.

[12] H. Zhuang, M. Yang, Z. Cui, and Q. Zheng, "A method for static hand gesture recognition based on non-negative matrix factorization and compressive sensing," *IAENG International Journal of Computer Science*, vol. 44, no. 1, pp. 52–59, 2017.

[13] Y.-S. Huang and Y.-J. Wang, "A hierarchical temporal memory based hand posture recognition method," *IAENG International Journal of Computer Science*, vol. 40, no. 2, pp. 87–93, 2013.

[14] Q. Zheng, X. Tian, S. Liu, M. Yang, H. Wang, and J. Yang, "Static hand gesture recognition based on gaussian mixture model and partial differential equation." *IAENG International Journal of Computer Science*, vol. 45, no. 4, pp. 569–583, 2018.

[15] H.-I. Suk, B.-K. Sin, and S.-W. Lee, "Hand gesture recognition based on dynamic bayesian network framework," *Pattern recognition*, vol. 43, no. 9, pp. 3059–3072, 2010.

[16] S. Mitra and T. Acharya, "Gesture recognition: A survey," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 37, no. 3, pp. 311–324, 2007.

[17] Q. Zhang, M. Yang, K. Kpalma, Q. Zheng, and X. Zhang, "Segmentation of hand posture against complex backgrounds based on saliency and skin colour detection." *IAENG International Journal of Computer Science*, vol. 45, no. 3, pp. 435–444, 2018.

[18] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 1, 2005, pp. 886–893.

[19] A. Klaser, M. Marszałek, and C. Schmid, "A spatio-temporal descriptor based on 3d-gradients," in *Proceedings of the British Machine Vision Conference 2008*, 2008.

[20] P. Scovanner, S. Ali, and M. Shah, "A 3-dimensional sift descriptor and its application to action recognition," in *Proceedings of the 15th ACM international conference on Multimedia*, 2007, pp. 357–360.

[21] I. Laptev, M. Marszalek, C. Schmid, and B. Rozenfeld, "Learning realistic human actions from movies," in *2008 IEEE Conference on Computer Vision and Pattern Recognition*, 2008, pp. 1–8.

[22] H. Wang and C. Schmid, "Action recognition with improved trajectories," in *Proceedings of the IEEE international conference on computer vision*, 2013, pp. 3551–3558.

[23] L. R. Rabiner, "A tutorial on hidden markov models and selected applications in speech recognition," *Proceedings of the IEEE*, vol. 77, no. 2, pp. 257–286, 1989.

[24] S. B. Wang, A. Quattoni, L.-P. Morency, D. Demirdjian, and T. Darrell, "Hidden conditional random fields for gesture recognition," in *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2, 2006, pp. 1521–1527.

[25] B. Liang and L. Zheng, "3d motion trail model based pyramid histograms of oriented gradient for action recognition," in *2014 22nd International Conference on Pattern Recognition*, 2014, pp. 1952–1957.

[26] G. Bishop, G. Welch *et al.*, "An introduction to the kalman filter," *Proc of SIGGRAPH, Course*, vol. 8, no. 27599-3175, p. 59, 2001.

[27] P. Hong, M. Turk, and T. S. Huang, "Gesture modeling and recognition using finite state machines," in *Proceedings Fourth IEEE International Conference on Automatic Face and Gesture Recognition*, 2000, pp. 410–415.

[28] M. Yeasin and S. Chaudhuri, "Visual understanding of dynamic hand gestures," *Pattern Recognition*, vol. 33, no. 11, pp. 1805–1817, 2000.

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

[30] A. Graves, A.-r. Mohamed, and G. Hinton, "Speech recognition with deep recurrent neural networks," in *2013 IEEE International Conference on Acoustics, Speech, and Signal Processing*, 2013, pp. 6645–6649.

[31] M.-T. Luong, I. Sutskever, Q. V. Le, O. Vinyals, and W. Zaremba, "Addressing the rare word problem in neural machine translation," *arXiv preprint arXiv:1410.8206*, 2014.

[32] W. Cao, A. Song, and J. Hu, "Stacked residual recurrent neural network with word weight for text classification." *IAENG International Journal of Computer Science*, vol. 44, no. 3, pp. 277–284, 2017.

[33] O. Vinyals, A. Toshev, S. Bengio, and D. Erhan, "Show and tell: A neural image caption generator," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 3156–3164.

[34] K. Soomro, A. R. Zamir, and M. Shah, "Ucf101: A dataset of 101 human actions classes from videos in the wild," *arXiv preprint arXiv:1212.0402*, 2012.

[35] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei, "Large-scale video classification with convolutional neural networks," in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2014, pp. 1725–1732.

[36] H. Kuehne, H. Jhuang, E. Garrote, T. Poggio, and T. Serre, "Hmdb: a large video database for human motion recognition," in *2011 International Conference on Computer Vision*, 2011, pp. 2556–2563.

[37] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[38] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[39] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[40] Q. Zheng, M. Yang, Q. Zhang, and J. Yang, "A bilinear multi-scale convolutional neural network for fine-grained object classification." *IAENG International Journal of Computer Science*, vol. 45, no. 2, pp. 340–352, 2018.

[41] Q. Zheng, M. Yang, Q. Zhang, and X. Zhang, "Fine-grained image classification based on the combination of artificial features and deep convolutional activation features," in *2017 IEEE/CIC International Conference on Communications in China (ICCC)*, 2017, pp. 1–6.

[42] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," in *Advances in neural information processing systems*, 2015, pp. 91–99.

[43] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," in *2017 IEEE International Conference on Computer Vision*, 2017, pp. 2980–2988.

[44] Q. Zheng, M. Yang, Q. Zhang, X. Zhang, and J. Yang, "An end-to-end image retrieval system based on gravitational field deep learning," in *2017 International Conference on Computer Systems, Electronics and Control (ICCSEC)*, 2017, pp. 936–940.

[45] K. Simonyan and A. Zisserman, "Two-stream convolutional networks for action recognition in videos," in *Advances in neural information processing systems*, 2014, pp. 568–576.

[46] C. Feichtenhofer, A. Pinz, and A. Zisserman, "Convolutional two-stream network fusion for video action recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 1933–1941.

[47] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri, "Learning spatiotemporal features with 3d convolutional networks," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 4489–4497.

[48] J. Konečnỳ and M. Hagara, "One-shot-learning gesture recognition using hog-hof features," *The Journal of Machine Learning Research*, vol. 15, no. 1, pp. 2513–2532, 2014.

[49] D. Wu, F. Zhu, and L. Shao, "One shot learning gesture recognition from rgbd images," in *2012 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, 2012, pp. 7–12.

[50] Y. M. Lui, "Human gesture recognition on product manifolds," *Journal of Machine Learning Research*, vol. 13, pp. 3297–3321, 2012.

[51] J. Wan, Q. Ruan, W. Li, G. An, and R. Zhao, "3d smosift: three-dimensional sparse motion scale invariant feature transform for activity recognition from rgb-d videos," *Journal of Electronic Imaging*, vol. 23, no. 2, p. 023017, 2014.

[52] X. Wang, M. Xia, H. Cai, Y. Gao, and C. Cattani, "Hidden-markov-models-based dynamic hand gesture recognition," *Mathematical Problems in Engineering*, p. 986134, 2012.

[53] J. Wan, G. Guo, and S. Z. Li, "Explore efficient local features from rgb-d data for one-shot learning gesture recognition," *IEEE transactions on pattern analysis and machine intelligence*, vol. 38, no. 8, pp. 1626–1639, 2016.

[54] O. Oreifej and Z. Liu, "Hon4d: Histogram of oriented 4d normals for activity recognition from depth sequences," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2013, pp. 716–723.

[55] S. Ji, W. Xu, M. Yang, and K. Yu, "3d convolutional neural networks for human action recognition," *IEEE transactions on pattern analysis and machine intelligence*, vol. 35, no. 1, pp. 221–231, 2013.

[56] L. Wang, Y. Xiong, Z. Wang, Y. Qiao, D. Lin, X. Tang, and L. Van Gool, "Temporal segment networks: Towards good practices for deep action recognition," in *European Conference on Computer Vision*, 2016, pp. 20–36.

[57] P. Molchanov, S. Gupta, K. Kim, and J. Kautz, "Hand gesture recognition with 3d convolutional neural networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, 2015, pp. 1–7.

[58] L. Pigou, A. Van Den Oord, S. Dieleman, M. Van Herreweghe, and J. Dambre, "Beyond temporal pooling: Recurrence and temporal convolutions for gesture recognition in video," *International Journal of Computer Vision*, vol. 126, no. 2-4, pp. 430–439, 2018.

[59] Y. Li, X. Wang, W. Liu, and B. Feng, “Deep attention network for joint hand gesture localization and recognition using static rgb-d images,” *Information Sciences*, vol. 441, pp. 66–78, 2018.

[60] A. Kurakin, Z. Zhang, and Z. Liu, “A real time system for dynamic hand gesture recognition with a depth sensor.” in *Proceedings of the 20th European Signal Processing Conference*, vol. 2, no. 5, 2012, p. 6.

[61] S. Ioffe and C. Szegedy, “Batch normalization: Accelerating deep network training by reducing internal covariate shift,” *arXiv preprint arXiv:1502.03167*, 2015.

[62] Q. Zheng, M. Yang, J. Yang, Q. Zhang, and X. Zhang, “Improvement of generalization ability of deep cnn via implicit regularization in two-stage training process,” *IEEE Access*, vol. 6, pp. 15 844–15 869, 2018.

[63] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, “Dropout: a simple way to prevent neural networks from overfitting,” *The Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[64] S. Azary and A. Savakis, “Grassmannian sparse representations and motion depth surfaces for 3d action recognition,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2013, pp. 492–499.

[65] H. Kumar and R. Ptucha, “Gesture recognition using active body parts and active difference signatures,” in *2015 IEEE International Conference on Image Processing*, 2015, pp. 2364–2368.

[66] J. Wang, Z. Liu, J. Chorowski, Z. Chen, and Y. Wu, “Robust 3d action recognition with random occupancy patterns,” in *2012 12th European Conference on Computer Vision*, 2012, pp. 872–885.

Yupeng Liu was born in Linyi, Shandong, China in 1994. He received B.S. degree from Shandong Agricultural University in 2016 and has been pursuing M.S. degree in Shandong University. His research interests include computer vision, deep learning and action recognition.