2021 International Conference on Technologies and Applications of Artificial Intelligence (TAAI) | 978-1-6654-0825-7/21/$31.00 ©2021 IEEE | DOI: 10.1109/TAAI54685.2021.00011

Trajectory-Based Dynamic Handwriting Recognition Using Fusion Neural Network

Tzu-An Huang*, Sai-Keung Wong† and Lan-Da Van‡
Department of Computer Science, National Yang Ming Chiao Tung University, Hsinchu, Taiwan
Email: *scotthuang0409@gmail.com, †cswingo@nycu.edu.tw, ‡ldvan@cs.nctu.edu.tw

***Abstract*—We propose a fusion network model for handwriting recognition. The model consists of a feedforward fully connected neural network (FNN) and a convolutional neural network (CNN). For a given handwriting trajectory, we generate two types of inputs for the FNN and CNN networks, respectively. Each of the networks produces a confidence vector for a handwriting trajectory. Subsequently, the fused result is the element-wise product of the two confidence vectors. We evaluated the proposed fusion network on two data sets, namely RTD and 6DMG, which contain alphabetic and numeric handwriting data. Five-fold cross validation was adopted. The average accuracy of our fusion network achieved 99.77% on the alphabetic data and 99.83% on the numeric data of the 6DMG data set, and 99.61% on the RTD data set. Finally, we compared the fusion network with three state-of-the-art techniques.**

***Index Terms*—handwriting recognition, feedforward fully connected neural network, convolutional neural network, fusion neural network**

I. Introduction

Hand gesture recognition provides an intuitive way to control using various hand gestures in a wide range of applications [1] [2]. A hand gesture recognition system has three stages, namely hand detection, hand gesture spotting, and hand gesture recognition. This paper focuses on the last stage. There are two types of hand gestures, namely static and dynamic gestures [3] [4]. A static hand gesture is a kind of static poses of hands, such as the OK sign and thumbs up [5], [6]. A dynamic hand gesture involves a hand motion consisting of a sequence of points that form a trajectory [7] [8]. Trajectory-based methods are popular in dynamic hand gesture recognition systems, which use the sequences of hand coordinates for recognition. Air handwriting recognition is a kind of dynamic hand gesture recognition for alphabets or numbers.

To recognize the dynamic hand gestures, Support Vector Machine (SVM) and Hidden Markov Model (HMM) [9] have been widely used [10] [11]. Yann et al. proposed a LeNet-5 convolutional neural network using a 28x28 handwriting character images for character recognition [12]. Nowadays, deep learning based methods become the main techniques for dynamic hand gesture recognition because of their high accuracy. Fusion neural networks learn multiple features and then combine these features for recognition. Hakim et al. [13] combined a long-short term memory (LSTM) and a 3D (dimensional) convolutional neural network (CNN) to learn temporal and spatial features, respectively. Molchanov et al. [7] used two 3D (dimensional) CNN models to extract the features. The final result is obtained by fusing the confidence vectors produced by the two CNNs based on element-wise multiplication.

In this paper, we propose a fusion network that uses a sequence of coordinates and trajectory images of alphabets and numbers as inputs of a fully-connected neural network (FNN) and a CNN, respectively. Colored strokes are used for capturing features of hand gesture trajectories, including directions and velocities. The FNN is used for learning temporal features of gesture trajectories and the CNN learns spatial features of trajectory images. We adopted the 5-fold cross validation scheme to evaluate the FNN, CNN, and fusion networks on two data sets, namely 6DMG [14] and RTD [15]. Our fusion network outperformed Alam's [15], Yana's [16], and Xu's [17] models.

II. Related Work

Singha et al. evaluated several classifiers for hand gestures, including the FNN, Support Vector Machine, k-th Nearest Neighbor, Naïve Bayes, and Extreme Learning Machine [18]. The result showed that the FNN achieved the highest accuracy compared to other classifiers. The CNNs have been employed for recognizing hand gestures because of their advantages on capturing spatial features of gesture trajectories [19] [20] [21] [22]. Hu et al. proposed to use a CNN which takes static trajectory images as inputs for dynamic hand gesture recognition [23]. Köpüklü et al. proposed to use a three-dimensional CNN for extracting the temporal features of hand gestures [24]. Alam et al. [15] proposed two architectures which have two LSTM layers [25] and a CNN for air handwriting recognition, respectively. Xu et al. [17] proposed a model to address unsupervised inertia-trajectory translation for character-level in air-handwriting. The main idea is that feature-level adversarial training is employed to the model. Thus, the model can map inertial and trajectory samples into semantic representations in a latent space.

Classifier fusion is a way to use two or more classifiers and combines the features or decisions for computing the final result. Singha et al. proposed a majority-voting method [11] to combine results obtained from several models for improving recognition performance. Molchanov et al. proposed an architecture consisting of two CNNs which take RGB-D hand images as inputs [7]. The final result is the element-wise product of multiple spatial scales obtained from the CNNs.

978-1-6654-0825-7/21/$31.00 ©2021 IEEE
DOI 10.1109/TAAI54685.2021.00011

Guillaume et al. combined two CNN models and a residual network model. The fused result is a combination of three computed features produced by the three models [8]. Hakim et al. [13] combined a 3D CNN and an LSTM to extract both spatial and temporal features, respectively. Then three different fusion models were evaluated for their performances [13]. Molchanov et al. proposed a model consisting of a 3D CNN and a recurrent neural network (RNN) [26]. Yana and Onoye proposed a fusion network using a CNN and a bidirectional LSTM [16].

III. Dataset

We employ two data sets, namely the 6DMG (6D Motion Gesture) and RTD (RealSense trajectory digits). They contain dynamic hand gestures. Figure 1 shows some examples of the two data sets.

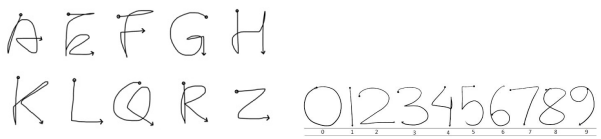


Fig. 1: Samples. Left: 6DMG. Right: RTD.

The 6DMG data set contains 6 DOF motion data and air handwriting data. It contains three kinds of data, motion gestures data, air-handwriting data, and air-finger writing data. Air-handwriting data set contains 600 numeric data and 6500 alphabetic data. They were written in a unique stroke. In the numeric data set, each number is repeated ten times with 6 participants. In the alphabetic data set, each alphabet is repeated ten times but with 25 participants. We only use the numeric data set and the upper case alphabets from "A" to "Z" in the air handwriting data set. The position data are used as input trajectories.

The RTD data set contains the air handwriting digits [15]. 20000 hand trajectories were collected through an Intel RealSense SR300 camera. Each trajectory consists of a sequence of three-dimensional coordinates. Let S be the sequence of the coordinates of a gesture trajectory. The data format of S is as follows:

$$S = \{(x_1, y_1, z_1), (x_2, y_2, z_2), \cdots, (x_n, y_n, z_n)\},$$

where n is the length of the gesture sequence. The z values are depth.

IV. Fusion neural network

Figure 2 shows our fusion neural network. The network consists of a feedforward FNN and a CNN. The architecture of CNN is based on Lenet-5 [12]. We need to determine the best combination of image input sizes, kernel sizes, and number of kernels because they affect greatly the recognition performance [27].

The entire process has two major stages, namely preprocessing and training. The preprocessing stage normalizes the data of the two data sets. Then it generates coordinate sequences and produces trajectory images. In the training stage, the networks are optimized based on a loss function. Finally, we evaluate the performance of the fusion network on testing data based on 5-fold cross validation.

A. Preprocessing

We normalize the position coordinates (x and y) of each data point of all trajectories to the interval $[0, 1]$. The depth-values of the data points are discarded. The FNN of our model takes an input sequence of coordinates of a hand gesture trajectory. Each input sequence has a fixed length ℓ which is set to 100. Thus, we use the nearest neighbor interpolation [28] to resample each trajectory sequence to obtain the corresponding input sequence.

We use the input sequences to generate the corresponding trajectory images. The input size of each trajectory image is m by m pixels. We adopt Algorithm 1 to rescale the coordinates of each point to the interval $[0, m-1]$. Here, we set m to 48. The following describes the purposes of Algorithm 1. **Lines 1 to 3:** Initialize variables. The padding space is required so that the strokes are contained inside the trajectory image. **Line 4:** Iterate all the gesture data. Process each of them, S. **Lines 5-14:** Determine the scaling factors, w' and h'. x_{min} and x_{max} are the minimum and maximum x-coordinates of S, respectively. y_{min} and y_{max} are the minimum and maximum y-coordinates of S, respectively. **Lines 15-18:** $\hat{x}_i$ and $\hat{y}_i$ are the normalized coordinates inside the interval $[0, m-1]$.

Algorithm 1: Normalization of trajectory coordinates

```
1  w = h = m;
2  p = 4 ;  padding size;
3  w_max = w − 2 ∗ p ;  max width;
4  for S ← get a trajectory from database do
5      Δx = x_max − x_min;
6      Δy = y_max − y_min;
7      α = Δx/Δy ;  aspect ratio;
8      if α > 1 then
9          w' = w_max/α;
10         h' = w_max;
11     else
12         w' = w_max;
13         h' = α w_max;
14     end
15     for i ← 1 to n do
16         x̂_i = (x_i − x_min) w'/Δx + w − w'/2;
17         ŷ_i = (y_i − y_min) h'/Δy + h − h'/2;
18     end
19 end
```

We describe how to produce a trajectory image for an input sequence as follows. At each data point of the input sequence we draw a colored stroke which is a square. The dimension of the colored stroke is 2x2 (Figure 3(a)). The colored stroke consists of blue, green, and red components. The blue and green components represent the spatial feature of a trajectory while the red component represents the temporal feature. The red component is computed as $r = \frac{i}{n}$, where i is the index of the data point and n is the length of the sequence.

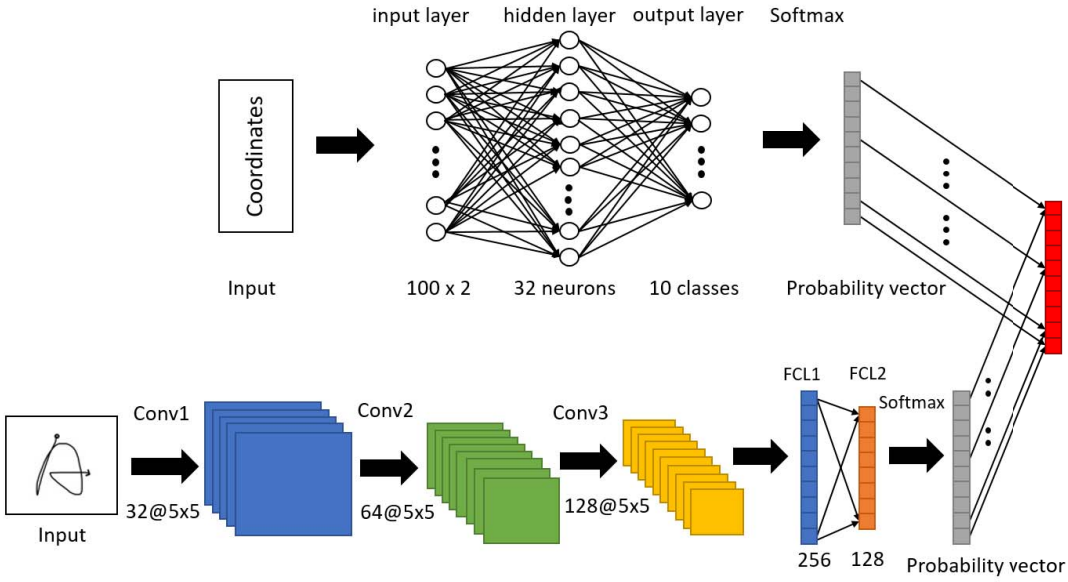


Fig. 2: Our fusion network architecture. The FNN accepts 100 sequence points and each point has two coordinates. In the CNN, the number of kernels in the first, second, and third convolutional layers are 32, 64, and 128, respectively. The number of kernels in the next layer is twice that of the previous layer. The kernel size of all convolutional layers is 5x5.

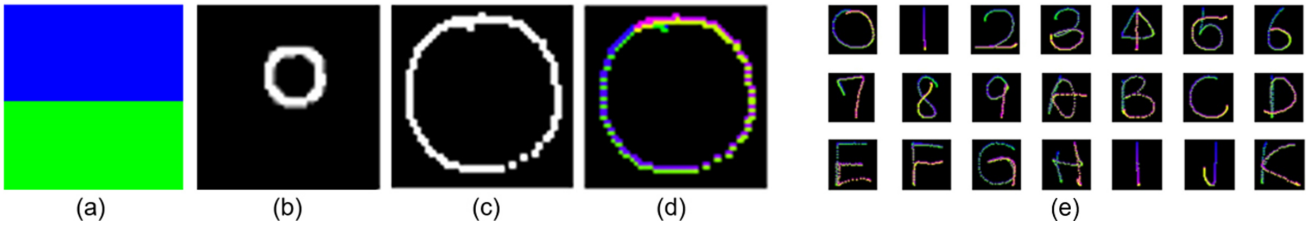


Fig. 3: (a) Colored stroke (green and blue). (b,c,d) Trajectory image of digit 0. (b) Original. (c) Normalized. (d) Normalized with colored stroke. (e) Colored trajectory images for some numbers and letters in 6DMG.

Figures 3(b,c,d) shows the original and normalized trajectory images for digit 0. The trajectory part that is drawn at the latter stage appears to be reddish or yellowish. The colored stroke is useful for representing directions and velocities of the respective trajectory. Figure 3(e) shows some examples.

In the 6DMG data set, we divide the data into four different types of data. The first three types are 1) numeric data, 2) alphabetic data, and 3) numeric and alphabetic data. The fourth type contains numeric and alphabetic data without digits 1 and 0, and letters I and O. These two pairs (i.e., (digit zero, letter O) and (digit one, letter I)) are ambiguous. Figure 4 shows that the trajectory images of letters I and O are almost the same as digits one and zero, respectively.

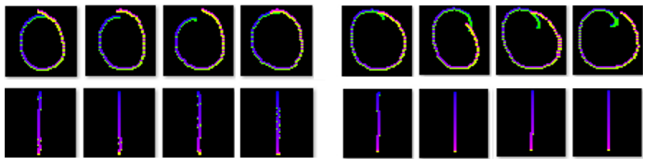


Fig. 4: Two pairs of ambiguous symbols in 6DMG. Left column: digits zero and one. Right column: letters O and I.

B. Training stage

The network parameters are updated to minimize the loss function which is the multiclass cross-entropy [29] [30]. Denote $L(W)$ as the loss function with parameters W (i.e., weights), N the number of the training sequences, and M the number of classes. $L(W)$ is defined as:

$$L(W) = -\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{M} log(P(c_j|x_i, W)),$$

where x_i is the gesture input and $P(c_j|x_i, W)$ is the probability for the data of the j-th class. We used Adam [31] to perform optimization on a single NVIDIA GeForce 1070

to train the networks. The batch size was 256 and the learning rate was 0.001. The dropout rate was 0.5 for preventing overfitting. We trained each neural network (FNN or CNN) for 200 epochs. Figure 5 shows the learning and accuracy curves. For the numeric data of 6DMG, the training times were 3.52 sec for FNN and 24.55 sec for CNN. For the alphabetic data of 6DMG, the training times were 15.88 sec for FNN and 192.99 sec for CNN. For the RTD data set, the training times were 35.35 sec for FNN and 726.75 sec for CNN. After the training stage, we proceeded to the testing stage to evaluate the performance of the networks. The testing time on the fusion network was about 3.0 ms per gesture trajectory on average.

V. Experiments and Results

We evaluated the performances of CNN structures with different combinations of input sizes, number of kernels, and kernel sizes. So that we could determine the best CNN structure. We trained twelve different CNN structures on the four types of data in the 6DMG data set. Table I shows the results. The accuracy for the numeric and alphabetic data is the lowest because of the two ambiguous pairs. However, the accuracy of the fusion result increases by up to 1.12% (see the setting (3x3, 16)). Thus, the FNN is useful to distinguish for the two ambiguous pairs. Furthermore, the results show that the fusion network indeed improves the overall accuracy. Table II shows the recognition accuracy on the RTD data set. The fusion network performs better than the CNN structures on average. Based on the results, we used 48^2 as the image size, and kernel size was 5x5. The number of kernels in the first convolution layer was 32 and the numbers of kernels in the next two layers were adjusted accordingly. We compared the fusion network with Alam's [15], Yana's [16], and Xu's [17] models. Table III shows that our fusion network outperforms all of them.

A. Performance Analysis of Our Networks

Table IV shows that the red component is useful. We compared the performance of the proposed fusion network with and without colored strokes, as shown in Table V. The CNN performs better for images with colored strokes than binary images on the three data sets (Alphabetic, Both, w/o ambiguous pairs) of 6DMG. This is because the colored strokes help encode the corner turning features of the gestures in the same class. However, the accuracy is higher for using binary images than images with colored strokes on the numeric data (98.83% in the two CNNs without/with the red channel).

Table VI shows the result on the RTD data set. As can be seen, the fusion results on the numeric and alphabetic data are little bit better for using images with colored strokes than binary images. However, we believe that the result is not reliable because of the two ambiguous pairs.

B. Discussion of fusion results

We analyze how the fusion network improves the overall performance. We collected some failure cases for the CNN but could be recognized correctly by the fusion network. CNN is good at learning spatial information, but the trajectory images of the same class vary. CNN may not handle special cases well. Figure 6 shows some samples of letters P and B in the 6DMG data set. Figure 7a shows a case that the CNN cannot recognize a letter B. The CNN recognizes the letter B as letter P because the vertical line occupies more region than that of most training data. However, the FNN focuses on the upper part of the trajectory, resulting in a higher confidence for this letter B. Figure 7b shows a case that the CNN cannot recognize a digit 9 in the RTD data set. The CNN recognizes the digit 9 as digit 0. The circle part of the digit 9 is larger than that of most training data and the second stroke is a straight line rather than a bit curved line. However, the FNN recognizes this digit 9 correctly.

VI. Conclusion

We propose a fusion network for handwriting recognition. The network combines a feedforward FNN and a CNN. The FNN and the CNN take a coordinate sequence and a trajectory image with color strokes as input, respectively. The color strokes implicitly encode the velocity, direction, and temporal features in the trajectory images. We evaluated the proposed fusion network on the 6DMG and RTD data sets. The experiment results showed that the accuracy of the proposed fusion network was higher than three state-of-the-art models which were Alam's [15], Yana's [16], and Xu's [17] models. For the RTD data set, the accuracy of our method was 99.61%. For the 6DMG data set, we achieved 99.83% accuracy on the numeric data set and 99.77% on the alphabetic data set.

Acknowledgment

We thanked the anonymous reviewers for their insightful comments. This project was supported in part by the Ministry of Science and Technology of the ROC under grant No. MOST 109-2221-E-009-121-MY3 and MOST PAIR LABs.

References

[1] D. Xu, "A neural network approach for hand gesture recognition in virtual reality driving training system of SPG," in *18th International Conference on Pattern Recognition*, vol. 3, 2006, pp. 519–522.

[2] S. Reifinger, F. Wallhoff, M. Ablassmeier, T. Poitschke, and G. Rigoll, "Static and dynamic hand-gesture recognition for augmented reality applications," in *International Conference on Human-Computer Interaction*. Springer, 2007, pp. 728–737.

[3] 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.

[4] J. S. Sonkusare, N. B. Chopade, R. Sor, and S. L. Tade, "A review on hand gesture recognition system," in *2015 International Conference on Computing Communication Control and Automation*, 2015, pp. 790–794.

[5] M. Z. Islam, M. S. Hossain, R. ul Islam, and K. Andersson, "Static hand gesture recognition using convolutional neural network with data augmentation," in *2019 Joint 8th International Conference on Informatics, Electronics & Vision and 2019 3rd International Conference on Imaging, Vision & Pattern Recognition*, 2019, pp. 324–329.

[6] R. F. Pinto, C. D. Borges, A. Almeida, and I. C. Paula, "Static hand gesture recognition based on convolutional neural networks," *Journal of Electrical and Computer Engineering*, vol. 5, pp. 1–12, 2019.

[7] 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.

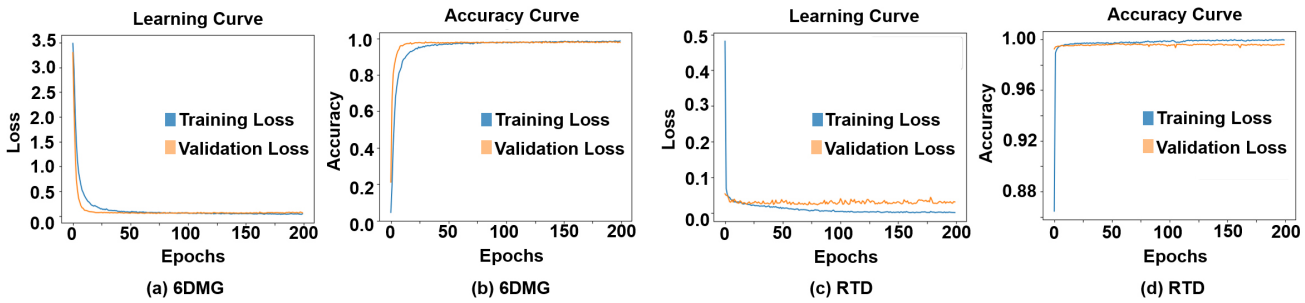


Fig. 5: Learning and accuracy curves at the training and validation stages. (a,b) 6DMG data set. (c,d) RTD data set.

CNN only

	Numeric	Alphabetic	Both	w/o 1,0,I,O
3x3,64	98.16%	99.55%	97.41%	98.83%
3x3,32	98.33%	99.46%	97.43%	98.91%
3x3,16	99.00%	99.49%	97.31%	98.93%
5x5,64	98.17%	99.38%	97.40%	99.01%
5x5,32	98.83%	99.48%	97.44%	98.87%
5x5,16	98.66%	99.46%	97.31%	98.67%

Fusion result

Numeric	Alphabetic	Both	w/o 1,0,I,O
99.67%	99.75%	98.84%	99.61%
99.33%	99.74%	98.76%	99.63%
99.50%	99.71%	98.93%	99.60%
99.33%	99.74%	98.86%	99.71%
99.67%	99.69%	98.76%	99.63%
99.33%	99.69%	98.86%	99.40%

Image size 32x32

	Numeric	Alphabetic	Both	w/o 1,0,I,O
3x3,64	98.16%	99.23%	97.96%	99.01%
3x3,32	97.50%	99.43%	97.82%	99.04%
3x3,16	98.00%	99.46%	97.75%	98.87%
5x5,64	98.67%	99.57%	98.24%	99.04%
5x5,32	98.83%	99.54%	98.11%	99.09%
5x5,16	98.67%	99.45%	97.83%	98.93%

Numeric	Alphabetic	Both	w/o 1,0,I,O
99.33%	99.69%	98.98%	99.55%
99.67%	99.64%	98.90%	99.72%
99.67%	99.66%	98.87%	99.72%
99.33%	99.67%	98.98%	99.68%
99.83%	99.77%	98.90%	99.75%
99.33%	99.75%	98.87%	99.71%

Image size 48x48

TABLE I: Accuracy results of the CNN structures and the fusion network on the 6DMG data set. Left column: kernel size, and number of kernels in the first convolution layer.

Image size 32x32

	CNN	Fusion
3x3,64	99.58%	99.63%
3x3,32	99.60%	99.63%
3x3,16	99.58%	99.62%
5x5,64	99.57%	99.62%
5x5,32	99.57%	99.62%
5x5,16	99.59%	99.63%

Image size 48x48

	CNN	Fusion
3x3,64	99.57%	99.61%
3x3,32	99.57%	99.62%
3x3,16	99.56%	99.61%
5x5,64	99.57%	99.62%
5x5,32	99.55%	99.61%
5x5,16	99.57%	99.61%

TABLE II: Results of the CNN structures and the fusion network on the RTD data set. Left column: kernel size, and number of kernels in the first convolution layer.

6DMG Dataset				
	Yana's [16]	Alam's [15]	Xu's [17]	Ours
Numeric	99.33%	99.32%	99.78%	99.83%
Alphabetic	99.27%		99.55%	99.77%
RTD Dataset				
Numeric	-	99.17%	-	99.61%

TABLE III: Comparison with other models on the 6DMG and RTD data sets.

	No red channel		Colored stroke	
Model	CNN	fusion	CNN	fusion
Numeric	98.83%	99.67%	98.83%	99.83%
Alphabetic	99.43%	99.74%	99.54%	99.77%
Both	97.16%	98.94%	98.11%	98.94%
w/o 1,0,I,O	98.39%	99.69%	99.09%	99.75%

TABLE IV: Recognition results for gestures with and without using the red component on the 6DMG data set.

[8] G. Devineau, F. Moutarde, W. Xi, and J. Yang, "Deep learning for hand gesture recognition on skeletal data," in *13th IEEE International Conference on Automatic Face & Gesture Recognition*, 2018, pp. 106–113.

[9] M. Awad and R. Khanna, "Hidden markov model," in *Efficient Learning Machines*. Springer, 2015, pp. 81–104.

[10] J. Singha and R. H. Laskar, "Ann-based hand gesture recognition using self co-articulated set of features," *IETE Journal of Research*, vol. 61, no. 6, pp. 597–608, 2015.

[11] J. Singha, S. Misra, and R. H. Laskar, "Effect of variation in gesticulation pattern in dynamic hand gesture recognition system," *Neurocomputing*, vol. 208, pp. 269–280, 2016.

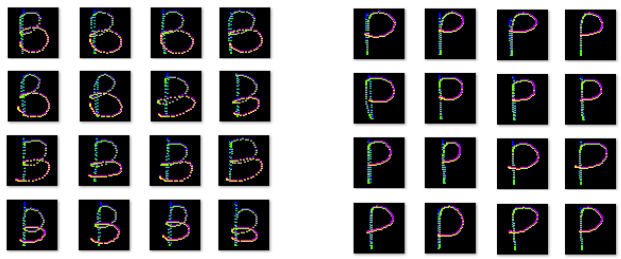
[12] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

	Binary images		Colored strokes	
Model	CNN	fusion	CNN	fusion
Numeric	98.67%	100.0%	98.83%	99.83%
Alphabetic	99.08%	99.72%	99.54%	99.77%
Both	96.89%	98.96%	98.11%	98.94%
w/o 1,0,I,O	98.39%	99.69%	99.09%	99.75%

TABLE V: Recognition results for binary images and images with color strokes on the 6DMG data set.

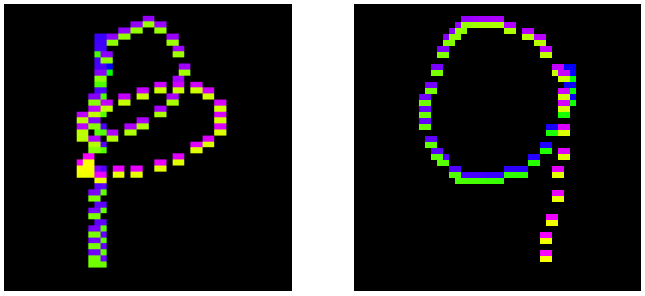
	Binary images		Colored strokes	
Model	CNN	fusion	CNN	fusion
Accuracy	99.57%	99.59%	99.57%	99.61%

TABLE VI: Recognition results for binary images and images with colored strokes on the RTD data set.



(a) Letter B (b) Letter P

Fig. 6: Trajectory images of letters B and P on 6DMG.



(a) Failure case for B (b) Failure case for 9

Fig. 7: Failure cases for B and 9 in the CNN. The CNN recognizes B as P and 9 as 0.

[13] N. L. Hakim, T. K. Shih, S. P. Kasthuri Arachchi, W. Aditya, Y.-C. Chen, and C.-Y. Lin, "Dynamic hand gesture recognition using 3dcnn and lstm with fsm context-aware model," *Sensors*, vol. 19, no. 24, p. 5429, 2019.

[14] M. Chen, G. AlRegib, and B.-H. Juang, "6dmg: A new 6d motion gesture database," in *Proceedings of the 3rd Multimedia Systems Conference*, 2012, pp. 83–88.

[15] M. Alam, K.-C. Kwon, M. Y. Abbass, S. M. Imtiaz, N. Kim *et al.*, "Trajectory-based air-writing recognition using deep neural network and depth sensor," *Sensors*, vol. 20, no. 2, p. 376, 2020.

[16] B. Yana and T. Onoye, "Fusion networks for air-writing recognition," in *International Conference on Multimedia Modeling*, 2018, pp. 142–152.

[17] S. Xu, Y. Xue, X. Zhang, and L. Jin, "A novel unsupervised domain adaptation method for inertia-trajectory translation of in-air handwriting," *Pattern Recognition*, vol. 116, p. 107939, 2021.

[18] J. Singha and R. H. Laskar, "Hand gesture recognition using two-level speed normalization, feature selection and classifier fusion," *Multimedia Systems*, vol. 23, no. 4, pp. 499–514, 2017.

[19] P. Roy, S. Ghosh, and U. Pal, "A cnn based framework for unistroke numeral recognition in air-writing," in *International Conference on Frontiers in Handwriting Recognition*, 2018, pp. 404–409.

[20] A. El-Sawy, M. Loey, and H. El-Bakry, "Arabic handwritten characters recognition using convolutional neural network," *WSEAS Transactions on Computer Research*, vol. 5, pp. 11–19, 2017.

[21] C. Yang, D. K. Han, and H. Ko, "Continuous hand gesture recognition based on trajectory shape information," *Pattern Recognition Letters*, vol. 99, pp. 39–47, 2017.

[22] S. Mukherjee, S. A. Ahmed, D. P. Dogra, S. Kar, and P. P. Roy, "Fingertip detection and tracking for recognition of air-writing in videos," *Expert Systems with Applications*, vol. 136, pp. 217–229, 2019.

[23] J.-T. Hu, C.-X. Fan, and Y. Ming, "Trajectory image based dynamic gesture recognition with convolutional neural networks," in *International Conference on Control, Automation and Systems*, 2015, pp. 1885–1889.

[24] O. Köpüklü, A. Gunduz, N. Kose, and G. Rigoll, "Real-time hand gesture detection and classification using convolutional neural networks," in *IEEE International Conference on Automatic Face & Gesture Recognition*, 2019, pp. 1–8.

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

[26] 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.

[27] W. S. Ahmed *et al.*, "The impact of filter size and number of filters on classification accuracy in cnn," in *International Conference on Computer Science and Software Engineering*, 2020, pp. 88–93.

[28] P. Molchanov, S. Gupta, K. Kim, and K. Pulli, "Multi-sensor system for driver's hand-gesture recognition," in *IEEE international conference and workshops on automatic face and gesture recognition*, vol. 1, 2015, pp. 1–8.

[29] C. M. Bishop, *Pattern recognition and machine learning*. Springer, 2006.

[30] S. Escalera, I. Guyon, and V. Athitsos, *Gesture recognition*. Springer, 2017.

[31] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.