Applied Intelligence (2020) 50:2239–2251
https://doi.org/10.1007/s10489-019-01607-0

Teach machine to learn: hand-drawn multi-symbol sketch recognition in one-shot

Chongyu Pan[1] · **Jian Huang**[1] · **Jianxing Gong**[1] · **Cheng Chen**[1]

Published online: 2 March 2020

Abstract
The ability to sequentially learn from few examples and re-utilize previous knowledge is an important milestone on the path to artificial general intelligence. In this paper, we propose Teach Machine to Learn (TML), a few-shot learning model for hand-drawn multi-symbol sketch recognition. The model decomposes multi-symbol sketch into stroke primitives and then explains the observed sequences in a bayesian criterion. A Bidirectional Long Short Term Memory (BiLSTM) encoder is employed for stroke primitives encoding. Meanwhile, a probabilistic Hidden Markov Model (HMM) is constructed for complete sketch inference and recognition. The challenging task of hand-drawn multi-symbol sketch recognition is implemented on two public datasets. The comparative results indicate that the proposed method outperforms the currently booming image-based deep models in recognition accuracy. Furthermore, our method is capable to continuously learn new concepts even in one-shot. The codes are currently available in https://github.com/chongyupan/Teach-Machine-to-Learn.

Keywords Multi-symbol sketch recognition · Few-shot learning · Lifelong learning · Probabilistic inference

1 Introduction

Artificial Intelligence (AI) has experienced three waves in processing information, especially in perceiving, learning, abstracting and reasoning abilities, according to DARPA's perspective. The first wave is known as handcrafted knowledge where reasoning about predefined problems is available without any learning capability and handling of uncertainty, such as the pre-programmed auto-pilot controller, the carefully designed Scale-Invariant Feature Transform(SIFT), Histogram of Oriented Gradients(HOG) image descriptors and other expert systems. After that, statistical learning methods are developed with the aid of huge amount of data and increasing computing power. Statistical models are constructed for special problems and trained on big data, such as the AlphaGo for chess playing [1], the popular Convolution Neural Networks (CNN) specialized for image feature representation [2]. However, challenges come about that they are statistically impressive but individually unreliable. Moreover, they are sensitive to be confused or cheated by skewed training data and slightly noisy samples [3, 4]. Furthermore, intelligence is much more about accumulation and use of knowledge for reasoning, planning, and learning than pattern recognition, even though it is mostly useful. This enlightens the third AI wave of contextual adaptation where contextual explanatory models are constructed for real world phenomenas.

End-to-end deep learning has achieved significant performance when recognizing some intractable source information, such as digital images, voice signals and natural languages. However, current deep models are heavily dependent on huge amount of labeled data and will forget previously incorporated data when trained with new ones, which is known as 'catastrophic forgetting'. Meanwhile, these supervised models must be taken off-line and retrained when encountering circumstances outside their training and cannot learn from sequential data adaptively.

To conquer the above drawbacks of the current deep learning methods, some innovative ideas spring up in a moment, such as few-shot learning [6, 7] that recognizes novel categories and concepts from very few training instances, lifelong learning [8, 9] that learns continuously

✉ Jian Huang
nudtjHuang@hotmail.com

Chongyu Pan
13548971657@163.com

[1] National University of Defense Technology, Changsha, China

during execution and applies previously obtained knowledge to novel situations.

Humans are able to learn new concepts from even one example and generalize well in later situations illustrated in Fig. 1. Inspired by compositionality, causality and learning to learn, three key principles in cognition science to support few-shot learning, a bayesian program and generative probabilistic model is proposed in [6]. It provides a promising way to learn from just several instances and adapt prior information to new concepts. Some new AI start-ups like Gamalon and Vicarious are also exploring the probabilistic generative models [10] towards more robust and general intelligence. To take a further step towards a more human-like and in-depth evaluation of an AI system, a recent work [11] even proposes the Kandinsky Patterns as IQ-test for machines, applying the principles of human intelligence tests.

Deep learning methods are naturally good at representing and encoding unstructured and nonlinear information while bayesian probabilistic methods are able to perform inference and reasoning with dynamic prior knowledge. As indicated in [12], major progress in Artificial General Intelligence (AGI) will come about through systems that combine representation learning with complex reasoning.

Specially, we propose a lifelong learning mechanism by teaching machine to learn structured symbols in one-shot. It will adaptively utilize the obtained knowledge for further multi-symbol sketch inference in real time. A neural network model is employed to encode hand-drawn stroke primitives and a probabilistic model is used for inference and online recognition based on the previously learnt symbols.

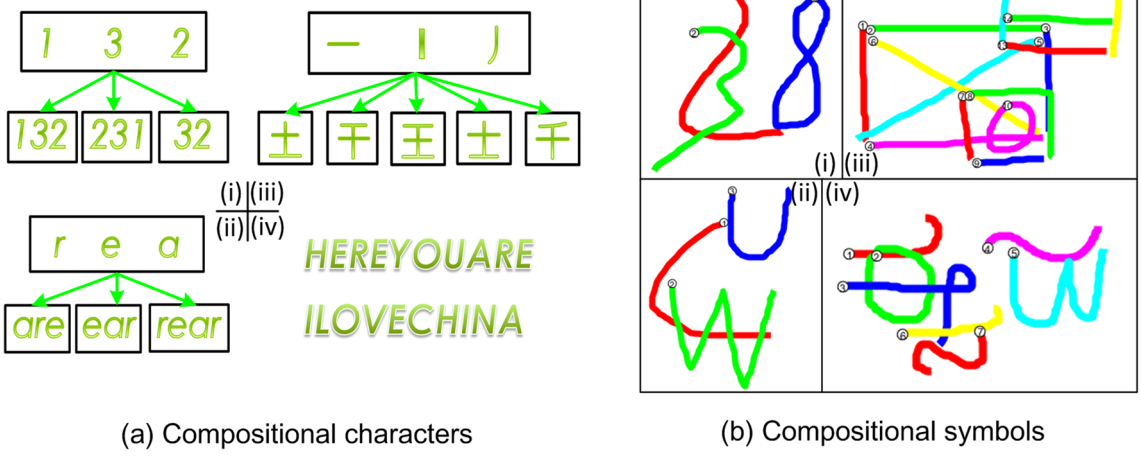
Although deep neural networks and probabilistic methods have been involved in many areas of machine learning, it is the first time to integrate both to the practical sketch recognition tasks, to our best knowledge. The contributions of this paper are summarized threefold:

- An effective and efficient teach-machine-to-learn mechanism is proposed to do few-shot learning and lifelong learning.
- A novel deep encoder-probabilistic inference framework is developed for sketch recognition.
- Extensive experiments on two practical datasets are conducted to validate the proposed method.

This paper is organized as follows. Some related works about sketch recognition are introduced in Section 2. Section 3 illustrates our framework of Teach Machine to Learn and the model details. After that, the experimental implementations on multi-symbol sketch recognition as well as the comparison results are given in Section 4. In Section 5, further analysises and discussions are presented. Finally, some conclusions are made in the last section.

2 Related works

Compared to traditional sophisticated WIMP(Window, Icon, Menu, Pointer) paradigm, hand-drawn sketching provides a more natural way for human-computer interaction [13]. The sketch-based interactions have been involved in many fields, such as sketch based image retrieval [14, 15], CAD drawings and military course of action diagrams [16, 17].



(a) Compositional characters

(b) Compositional symbols

Fig. 1 Humans are able to continuously learn new concepts via few-shot examples by three principles: compositionality, causality and learning to learn. **a** Different kinds of characters are composed of limited symbol primitives: (i) Arabic numerals; (ii) English letters; (iii) Chinese character strokes. (iv) One is able to infer and recognize the continuous sentence based on the individual vocabularies. **b** Segmenting multi-symbol sketch into individual symbols and parsing each symbol into primitive strokes: (i) multi-digit sketch; (ii) multi-letter sketch; (iii) multiple military symbols from COAD [5]; (iv) multiple characters from Omniglot [6]

A large-scale empirical study [6] about sketch drawing revealed that different drawers tend to be highly consistent in the shape, order and direction of the pen strokes when constructing a particular symbol. So it is possible for machine to automatically recognize the hand-drawn sketch by modeling the drawing procedure. In spite of the dynamics of the motor programs shared for a same hand-drawn symbol, automatic sketch recognition still remains a critical and challenging problem due to the varying drawing styles, the similarity across symbols and even the uncertain number of symbols in a complete sketch.

A great deal of attentions have been paid to hand-draw sketch recognition for a long time. Early works mainly focused on the temporal strokes information or spatial image appearance by handcrafted features. A typical system named PaleoSketch recognizer [18] classifies eight determinate primitives based on strokes information, such as the 'normalized distance between direction extremes' and 'direction change ratio' features. Other features such as convex hull, perimeter, area scalar ratios [19], direction, curvature, speed for strokes [20] and even Fourier descriptors [21] are also used to recognize hand-draw shapes, strokes and trajectories.

Another way to recognize sketch is to extract image features and match with predefined templates via various image descriptors, including HOG [22], Image Deformation Model (IDM) [5, 23], Self-Similarity (SSIM) [24], Shape Context (SC) [25, 26], Fisher vectors [27] and so on. In addition to only relying on single feature descriptor, some attempts fusing multiple complementary features are exploited to tackle the visual cue sparsity problem [28]. Especially, a Multiple Kernel Learning(MKL) model [29] was proposed to fuse multiple features and even similarity metrics, achieving perfect performance among image feature based solutions.

Most relevant works in literature focused on the single-symbol sketch recognition where an isolated symbol is detected and classified. Only a few attentions were paid to continuous sketch recognition demonstrated in Fig. 1b where the scene may contain several sequential symbols. A step-by-step method [30] recognizes the complete sketch in special domains. It divides strokes into segments at the detected corner points and then recognizes generated candidate symbols by IDM feature matching. Another holistic solution [31, 32] handles the segmentation and classification of the complete sketch simultaneously by maximizing the joint likelihood of multi-symbol patterns using HMM model. However, the primitives representation is simply based on line segments. It is difficult to generalize to multi-symbol sketch recognition where more complicated stroke primitives need to be encoded and recognized in a flexible way.

Recently there has been a surge of interest in scene sketch recognition and understanding. Several large-scale datasets are designed [33–35] for object-level or scene-level sketch understanding and provide foundational benchmarks for follow-up studies. Some special neural network models [36, 37] are even designed to investigate recognition or semantic segmentation of scene sketches. Sun et al. [38] developes a sketch segmentation framework and recognizes sketches with the aid of one million annotated clipart images. Based on the co-occurrence and relation knowledge learned from extra scene images, [39] intends to classify a sketched object by considering its surrounding context. In addition to basic sketch recognition task, some researches [40, 41] even focus on zero-shot image retrieval based on hand-drawn sketches. These works are almost based on image pixels and using handcrafted features or learning deep feature representations. Different from the above mentioned works, Google developes a drawing application named $quickly, draw!$ [33] which aims to learn and recognize general concepts in a manner like humans. A Recurrent Neural Network (RNN) $sketch-rnn$ is implemented to construct stroke-based drawings of common objects at the back end. Inverse to the sketch-to-photo retrieval, [42] addresses the photo-to-sketch problem using a learned deep model for the first time. Combining spatial CNN features and temporal RNN features, [43] proposes a two-branch CNN-RNN deep sketch hashing network for large scale human sketch retrieval.

Although great progresses have been made, the current works tackle the problem with the aid of sufficient well-annotated sketch data. They are ineffective to generalize to data-scarcity cases, and even unable to work in a continuous learning way. Inspired by humans being able to learn new concepts from limited data, the Bayesian Program Learning (BPL) [6] approach defines a generative model by learning probabilistic programs to represent concepts and building them hierarchically from parts, subparts and spatial relations. Although data efficient, it is heavily dependent on statistical prior probability information. After that, another impressive work introduces a probabilistic generative model named Recursive Cortical Network (RCN) [10] that breaks the challenging CAPTCHAs tasks. It presents robust generalization and reasoning abilities on scene text recognition tasks with much more data efficiency than deep learning methods.

Combining the powerful representing ability of deep learning method and inference capability of probabilistic method, we intend to teach machine to learn symbols by encoding primitive strokes and modeling spatial relationships. Based on the obtained single symbol knowledge, our method is able to segment, infer and recognize the continuously drawn multi-symbol sketch by maximizing a posterior probability of the constructed hidden markov model. The evaluations on two special datasets show that it provides compelling performance in varying situations than

currently prevalent deep models while being simpler and more flexible than the alternatives.

3 Teach machine to learn

3.1 Model architecture

Figure 2 shows the three-level model architecture that represents the composition process of the multi-symbol sketch. The hand-drawn sketch consists of multiple sequentially drawn symbols and each individual symbol is made up of separate primitive strokes. Each sketch is collected as a set of pen stroke points in drawing pad. Each point is formulated as (I_x, I_y, p) where (I_x, I_y) is the initial 2D absolute coordinate and p indicates a binary state whether the point is the last one in a stroke. The binary indicator p is determined by judging that the pen is persistently touching the pad or will be lifted from the pad after the point. Individual strokes are then separated by indicator p and n sample points are collected for each stroke by equivalent interval sampling. Each stroke S_i is represented as a sequence of sample points $(\boldsymbol{x_1}, \boldsymbol{x_2}, \cdots, \boldsymbol{x_n})$ where $\boldsymbol{x_i} = (\delta I_x, \delta I_y)$ is the normalized offset coordinate of the current sample point from the previous one.

3.2 Primitives encoding and spatial relationship modeling

Primitives encoding. When drawing an identical stroke, people may have different style preferences and inconsistent stroke directions, such as a slope line drawn from the bottom up or top down. Considering the varying drawing styles and bidirectional stroke directions, we exploit a BiLSTM encoder for basic primitives recognition. It takes in a point sequence $(\boldsymbol{x_1}, \boldsymbol{x_2}, \cdots, \boldsymbol{x_n})$ as input, and outputs a classification vector predicting its category attribute A_i. The BiLSTM model architecture is detailed in Fig. 3b, where each block represents a basic Long Short Term Memory(LSTM) cell depicted in Fig. 3a and formulated as following:

$$\begin{aligned}
\boldsymbol{f_t} &= \sigma(\boldsymbol{W_f x_t} + \boldsymbol{U_f h_{t-1}} + \boldsymbol{b_f}) \\
\boldsymbol{i_t} &= \sigma(\boldsymbol{W_i x_t} + \boldsymbol{U_i h_{t-1}} + \boldsymbol{b_i}) \\
\boldsymbol{o_t} &= \sigma(\boldsymbol{W_o x_t} + \boldsymbol{U_o h_{t-1}} + \boldsymbol{b_o}) \\
\boldsymbol{c_t} &= \boldsymbol{f_t} \circ \boldsymbol{c_{t-1}} + \boldsymbol{i_t} \circ \tanh(\boldsymbol{W_c x_t} + \boldsymbol{U_c h_{t-1}} + \boldsymbol{b_c}) \\
\boldsymbol{h_t} &= \boldsymbol{o_t} \circ \tanh(\boldsymbol{c_t}) \\
\tanh(x) &= \frac{e^x - e^{-x}}{e^x + e^{-x}}
\end{aligned} \tag{1}$$

where $\boldsymbol{x_t}, \boldsymbol{c_t}, \boldsymbol{h_t}$ are the input, internal memory and hidden state at time step t. $\boldsymbol{i_t}$, $\boldsymbol{f_t}$, $\boldsymbol{o_t}$ denote the input, forget and output gates, respectively. All the $\boldsymbol{W}$, $\boldsymbol{U}$ and $\boldsymbol{b}$ are corresponding parameter matrixes remained to be learnt in the training sessions. σ is the sigmoid function and $\circ$ is an operator denoting element-wise multiplication.

After n step iterations with the input sequence $(\boldsymbol{x_1}, \boldsymbol{x_2}, \cdots, \boldsymbol{x_n})$, the output of the BiLSTM encoder is finally combined as

$$\begin{aligned}
\boldsymbol{y_i} &= softmax(\boldsymbol{W_y}[\boldsymbol{h_n^B}, \boldsymbol{h_n^F}] + \boldsymbol{b_y}) \\
A_i &= \underset{j}{\operatorname{argmax}} \boldsymbol{y_{ij}}
\end{aligned} \tag{2}$$

where $[\boldsymbol{h_n^B}, \boldsymbol{h_n^F}]$ denotes the concatenation of the two vectors $\boldsymbol{h_n^B}$ and $\boldsymbol{h_n^F}$. $\boldsymbol{W_y}$ and $\boldsymbol{b_y}$ are the parameters. The $softmax$ is a normalization function that converts a vector to a normalized probability distribution for the next classification step. The category attribute A_i is the index corresponding to the maximum component in the classification vector $\boldsymbol{y_i}$.

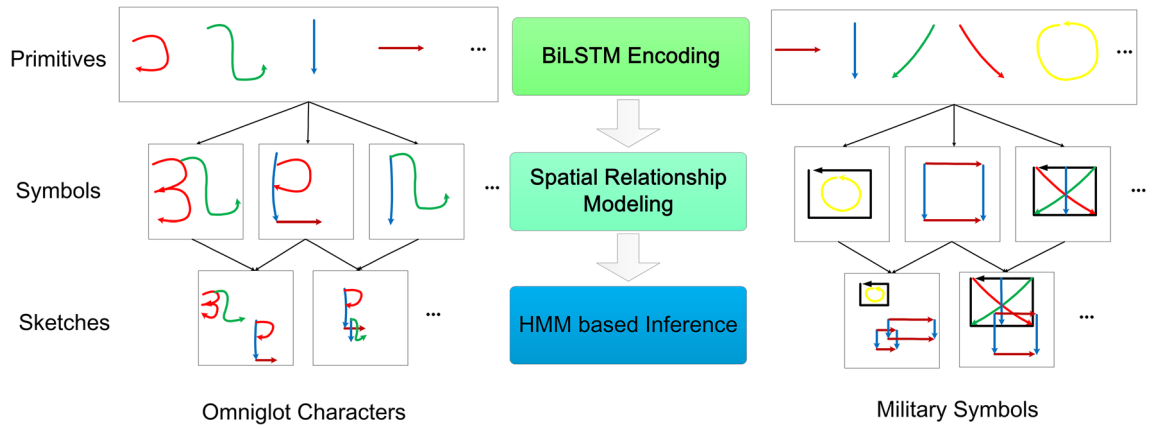


Fig. 2 Architecture overview for multi-symbol sketch recognition: sketches are made up of multiple symbols and individual symbols consist of several sequential primitive strokes with spatial relation constrains

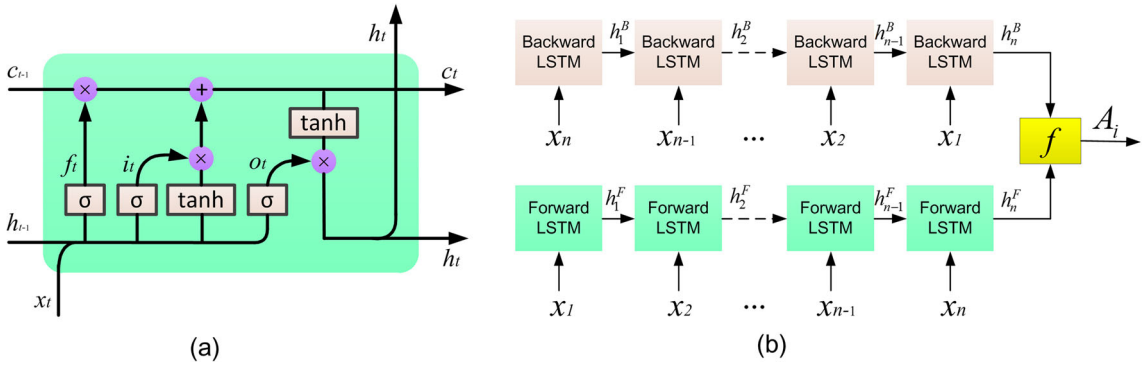


Fig. 3 **a** LSTM cell structure and **b** BiLSTM model used for encoding primitive strokes

Spatial relationship modeling. The spatial relationship R_i indicates how each stroke S_i connects to its previous stroke S_{i-1}. According to the starting position of the current stroke, relationships come in four types, including 'connected at start', 'connected at end', 'connected at middle' and 'no connection' illustrated in detail in Fig. 4. As shown in Fig. 4, the connection is admitted valid only if two points keep apart within a predefined distance threshold.

3.3 HMM based inference

Sketch representation. Based on the stroke encoding and spatial relationship modeling, a l-stroke sketch will be represented as $(S_1, S_2, \cdots, S_l)$ where $S_i = (A_i, R_i)$. A_i and R_i represent the attribute and the spatial relationship of the i-th stroke, respectively.

Online learning. The first step is to teach machine to learn new single symbol even in one-shot. A supervised single symbol is provided by user and the machine takes the recognized result as a template $Sym_i = (S_1, S_2, \cdots, S_l)$. With the one-by-one symbol teaching and online learning mechanism, it is flexible to add, delete or substitute symbols to a dynamically changing template library $\{Sym_i, i=1, 2, \cdots, m\}$ as users need. Meanwhile, the template library could be pre-defined offline before real-time operation.

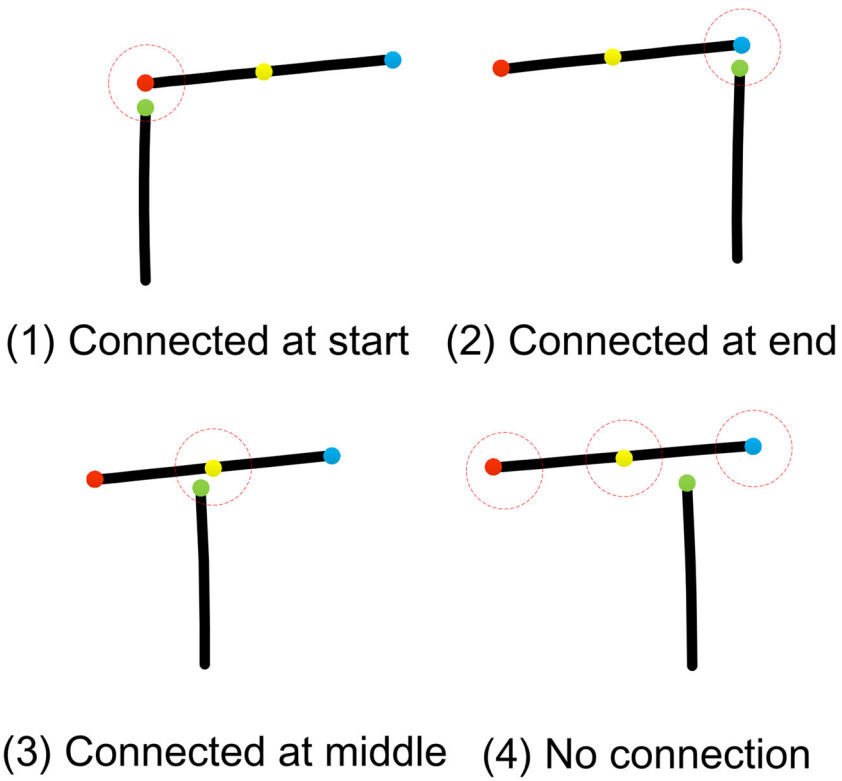


Fig. 4 Four kinds of spatial relationships between a stroke and the previous one: the stroke S_i (the vertical line) starts at the (1) beginning, (2) end, (3) middle of or (4) does not connect to the previous stroke S_{i-1} (the horizontal line)

Online Recognition. For a test multi-symbol sketch $Sk^T = (S_1, S_2, \cdots, S_l)$ and already constructed template library $\{Sym_i, i=1, 2, \cdots, m\}$, the goal is to segment Sk^T into k individual symbols $\{Sk_i, i=1, 2, \cdots, k\}$ and label each symbol Sk_i with a template Sym_{y_i}. We take a bayesian perspective that maximizes the posterior probability

$$\underset{k, y_1, y_2, \cdots, y_k}{\text{argmax}} \; P(Sk_1, Sk_2, \cdots, Sk_k | Sym_{y_1}, Sym_{y_2}, \ldots, Sym_{y_k}) \tag{3}$$

where k is the number of the individual symbols for the test sketch and y_k is the corresponding index in the template library.

We solve this maximization problem by modeling stroke sequences as a HMM chain and inferring the segmentation results using dynamic programming and the shortest path algorithm.

As shown in Fig. 5a, given an observation sequence $(S_1, S_2, \cdots, S_l)$ of the test sketch, we first construct a HMM chain of l+1 nodes. Each node represents a stroke $S_i = (A_i, R_i)$ and the last one indicates the end of the chain.

Then we are going to assign hypothesis symbol templates to subsequences of the observation sequence $(S_1, S_2, \cdots, S_l)$ to segment and infer the test sketch. Taking the first template Sym_1 as a sample, a directed edge from S_1 to S_{r+1} (r is the strokes number of the current template) is added with an associated cost of

$$-lg\, P((S_1, S_2, \ldots, S_r) | Sym_1) \tag{4}$$

and the similarity probability is defined as

$$\begin{aligned} P(Sym_i | Sym_j) &= \frac{1}{2r}(A_{ij} + R_{ij}) \\ A_{ij} &= \sum_{k=1}^{r} A(i,k) == A(j,k) \\ R_{ij} &= \sum_{k=1}^{r} R(i,k) == R(j,k) \end{aligned} \tag{5}$$

where Sym_i , Sym_j are two stroke sequences with r strokes each and $A(i, k)$, $R(i, k)$ are the attribute and relationship of the k-th stroke in Sym_i, respectively.

Similar to the case above, the candidate template Sym_1 is matched with the remaining subsequences $(S_2, S_3, \cdots, S_{r+2})$, $(S_3, S_4, \cdots, S_{r+3})$ until the last one $(S_{l+1-r}, S_{l+2-r}, \cdots, S_l, END)$. The corresponding edges with associated costs are established following (4).

Following Sym_1, we keep assigning each template to the observation sequence in a recursive way until covering all the existing symbols in the template library, as shown in Fig. 5b.

In Fig. 5b, a directed edge from S_i to S_{r+i} with cost c indicates that it is able to account for the observation subsequence $(S_i, S_{i+1}, \cdots, S_{r+i-1})$ with the r-stroke symbol with a log likelihood of $-c$. There may be multiple edges in the constructed graph connecting two nodes, corresponding to alternative symbol templates and costs.

Given the constructed graph, we are going to find a sequence of hypotheses which accounts for the whole observation sequence without gaps or overlaps, such that

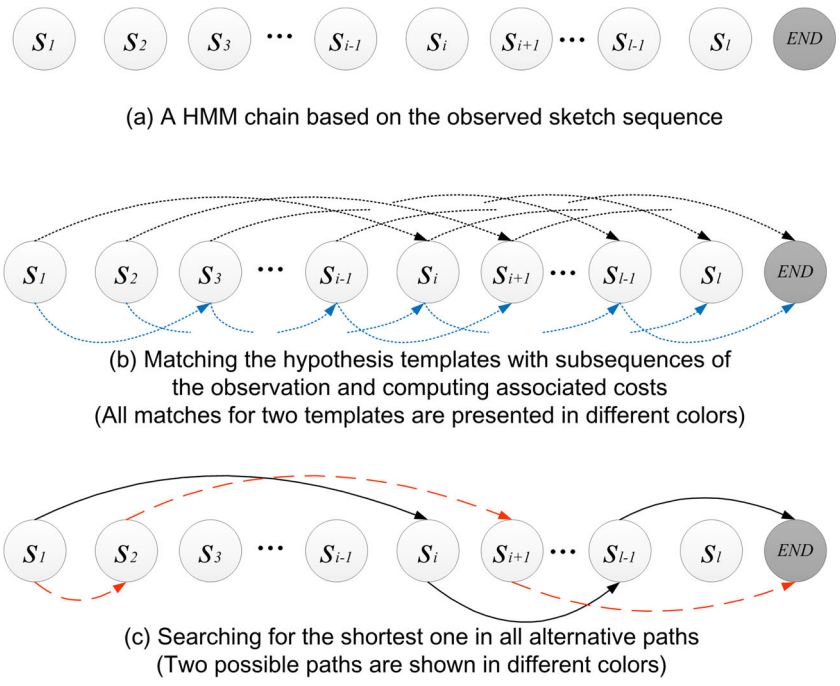


Fig. 5 Multi-symbol sketch inference based on HMM model

the sum of the costs for the hypotheses is minimum, as shown in Fig. 5c. By searching the shortest path from S_1 to END using the dynamic programming algorithm, we minimize sum of the negative log likelihoods, equivalent to maximizing the posterior probability in (3). The indices of the shortest path demonstrate the segmentation of the multi-symbol sketch and the symbols are identified by the hypothesis templates corresponding to each segments.

4 Experiments

In this section, we describe two sketch datasets in detail and present the experimental results. Specifically, we compare the TML method against prevailing baselines and validate the few-shot learning ability.

4.1 Datasets and settings

Datasets COAD (Course of Action Diagrams) dataset [5] is a subset of hundreds of US military symbols used for field operation planning. It contains 20 categories and some symbols have distinctive shapes whereas others appear as a part of one or more symbols. Firstly, we collect 2000 single symbols with 100 samples per category by 10 participants. Figure 6 shows the primitives for BiLSTM encoder training, user-drawn single symbols and multi-symbol sketch test data. It is worth to note that multiple symbols will inevitably get overlapping with each other and embedded in Common Operational Picture (COP) background in practical applications. For this sake, we generate multi-symbol sketch test data by overlapping randomly sampled symbols. As shown in Fig. 6c, the single symbols are overlapped in horizontal direction with different overlap rate (defined as the ratio of overlap width to symbol width). Specifically, for each experiment episode, 1100 multi-symbol sketches are randomly generated with 100 examples per overlap level (11 levels of overlap rate from 0.0 to 1.0).

Omniglot [44] is a public few-shot learning benchmark dataset with 105*105 binary images as well as stroke-by-stroke drawing information. Unlike MNIST which has a small number of classes with thousands of examples each, Omniglot dataset contains 1623 character categories from 50 different alphabets with only 20 hand-drawn examples per category. Some symbols from different categories look similar to each other while there may be obvious variations in stroke number, order and style within the same category. With high inter-class similarity and intra-class diversity, the dataset is challenging for machine and even for human judges. Figure 7 presents some of the Omniglot characters, 17 summarized primitive strokes for BiLSTM encoder training and multi-symbol sketch test data. Similarly, the test sketches are generated by overlapping randomly sampled symbols in horizontal direction with various overlap rate.

Settings About the stroke primitives encoding, the number of sample points for each stroke is set as n=10 for COAD symbols and 20 for Omniglot symbols. For COAD dataset,

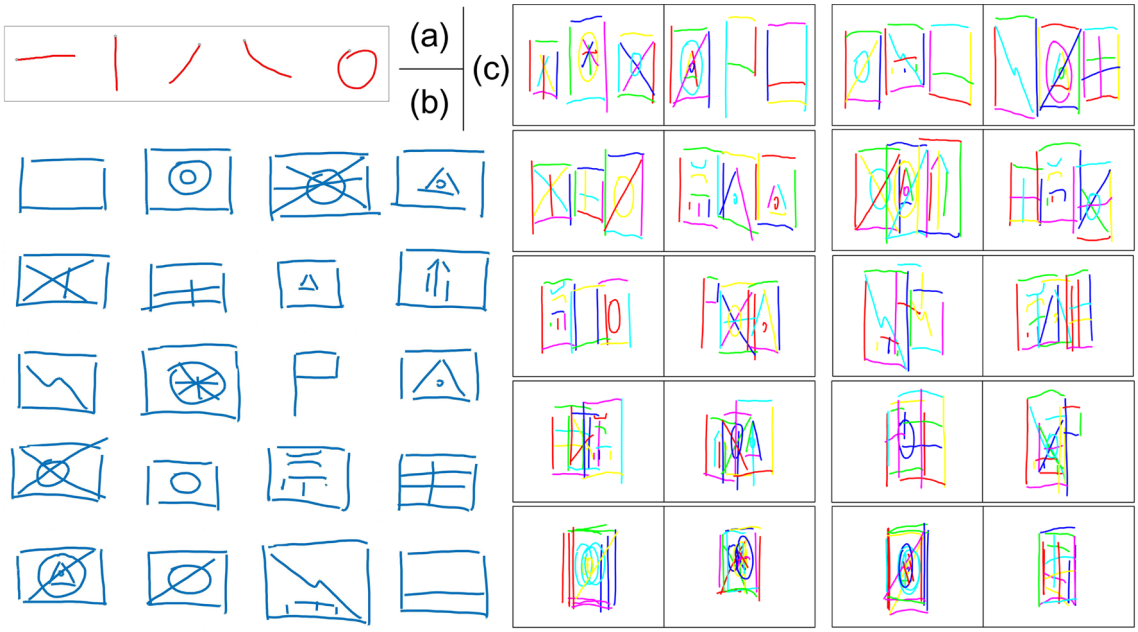


Fig. 6 COAD dataset. **a** 5 primitives to encode strokes. **b** 20 symbols. **c** Test data: From top left to bottom right, the presented sketch sets (2 samples per set) are generated with overlap rate being 0.1, 0.2, $\cdots$, 1.0

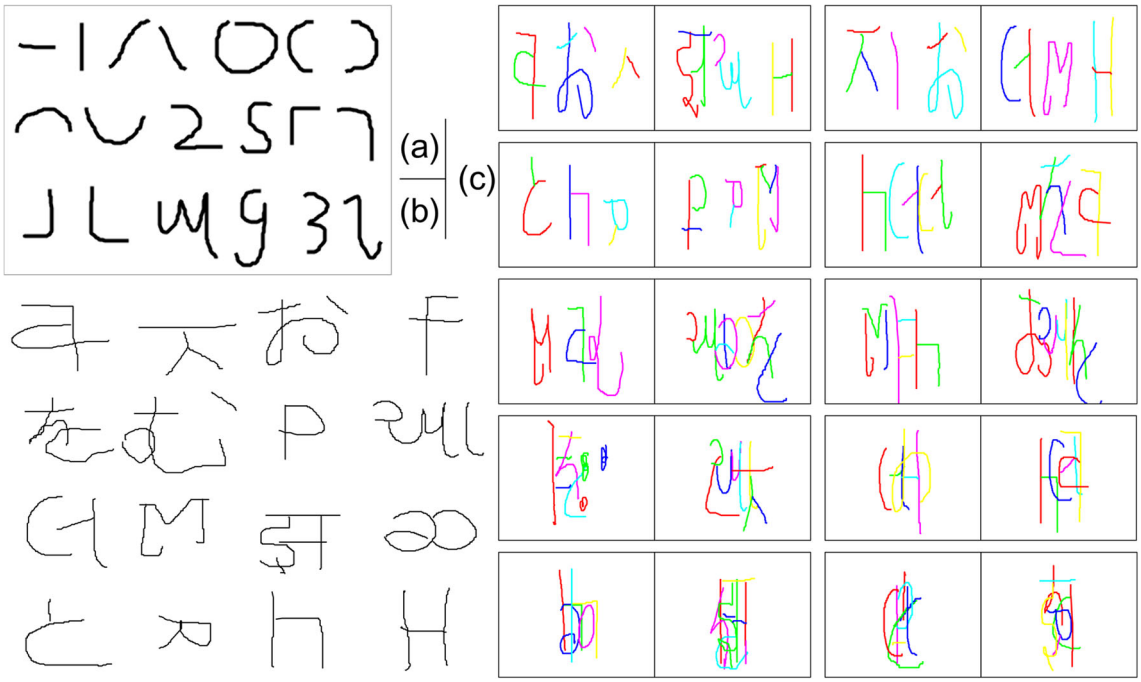


Fig. 7 Omniglot dataset. **a** 17 primitives to encode strokes. **b** 16 (out of 1623) characters. **c** Test data: From top left to bottom right, the presented sketch sets (2 samples per set) are generated with overlap rate being 0.1, 0.2, $\cdots$, 1.0

we provide only 5 samples per category to construct the template library while 90 samples per class are prepared for other baseline methods. For Omniglot dataset, we randomly select 20 classes in 1623 characters for performance comparison.

Evaluations The experiments are carried out on multi-symbol sketches with different overlap rate as illustrated in Figs. 6c and 7c. For multi-symbol sketch recognition, correctly recognized means that all the individual symbols in the sketch are properly segmented and labeled with the ground-truth tags. The recognition accuracy is measured by $Acc = N_c \ / \ N_t$ where N_c and N_t denote the number of the correctly recognized samples and the total samples, respectively. The final results are recorded by averaging over 10 randomly generated episodes for the proposed TML method.

4.2 Comparison with baseline methods

Baselines For comprehensive and sufficient comparison, several significant methods in object detection domain are employed for multi-symbol sketch recognition. For example, the widely used Region-based Convolutional Neural Networks(RCNN) [45], You Only Look Once(YOLO) [46], Single Shot Detector(SSD) [47] and even more recent Feature Pyramid Network(FPN) [48] are included as baselines. As a fundamental method for object detection, RCNN consists of three procedures, including region proposal, feature extraction and binary linear SVM classification. We reproduce this baseline method with step-by-step implementation. The multi-symbol sketch images are used here as domain-specific training data. However, the randomly sampled symbols in these sketches are arranged in horizontal direction without overlapping. For feature extraction, we finetune the pre-trained Alexnet [2] (pre-trained on the ImageNet dataset) with the multi-symbol sketch images. All region proposals $\geq$ 0.5 IOU (intersection-over-union) overlap with the ground-truth box are regarded as positive samples and the rest as negatives. For SVM classifiers, only the ground-truth boxes are positive examples while proposals $<$ 0.3 IOU overlap are defined negatives. When recognizing a multi-symbol sketch image, we randomly sample 2000 region proposals on the test image and extract their 4096-dimensional FC7 (the 7-th fully connection layer in Alexnet) feature vectors to binary SVM classifiers. The classifier categories with top N votes (N is the ground-truth symbols number in the test sketch) are regarded as the final recognition results. For another three baselines including YOLO, SSD and FPN, we implement the training and evaluation procedures using the original published codes as well as the parameter settings.

Results The results are shown in Table 1 where our TML method and other baselines are evaluated on COAD and Omniglot datasets, respectively. To further visualize the

Table 1 Comparison results of several baselines and our proposed method

	Overlap rate	0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
COAD	RCNN	90	99	98	79	41	25	5	4	4	3	5
	YOLO	35	32	36	43	27	26	26	26	25	12	18
	SSD	12	16	15	17	14	13	18	15	13	8	10
	FPN	32	31	33	30	44	39	40	34	23	22	27
	TML	80	83	83	82	83	84	81	82	82	83	83
Omniglot	RCNN	63	68	74	46	29	17	15	7	9	4	4
	YOLO	43	28	29	46	40	43	35	34	39	31	30
	SSD	15	13	10	13	20	20	18	21	19	27	23
	FPN	36	29	34	37	19	34	26	20	21	21	23
	TML	86	84	84	84	84	84	86	86	86	85	85

The multi-symbol sketch recognition accuracies are listed in %

accuracy varying trend with gradually overlapping, we depict the recognition accuracy versus overlap rate in Fig. 8. It is obvious that the accuracy of step-by-step RCNN is going lower overall with the increasing overlap rate on both datasets while TML keeps strikingly stable with almost no disturbance. A further look at the figure, we could find that RCNN is nearly perfect for slight overlap on COAD (The accuracy reaches 99% when overlap rate is 0.1) where the multiple symbols have not been stacked in raw images as shown in Fig. 6c, indicating that the deep learning methods are naturally good at representing pre-trained knowledge but brittle and powerless to handle varying situations. In terms of other methods, such as YOLO, SSD and FPN, the accuracies keep in a low level due to the data sparsity and overfitting optimization problem, although the top-1 accuracy is utilized as the evaluation criteria in these cases. In contrast, our robust TML method performs better with 82.31% average accuracy on COAD and 84.75% on Omniglot, based on fewer samples. Furthermore, with lifelong learning mechanism, it is able and flexible to continuously learn more concepts online as users like.

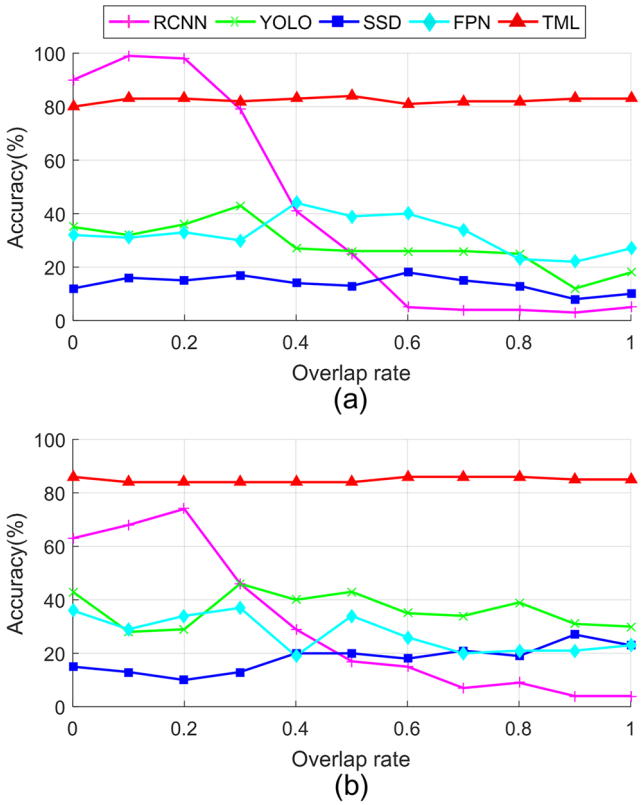


Fig. 8 Multi-symbol sketch recognition accuracy on (**a**) COAD and (**b**) Omniglot

4.3 Few-shot recognition

Few-shot learning aims to recognize novel concepts or categories from very few labeled examples. To see how the model performs for few-shot learning, we implement C-way K-shot experiments on COAD dataset. By C-way K-shot, we mean that the test set is composed of C-category symbols and only K labeled examples per category are given to train the model. Following the standard settings used in most related few-shot learning works [7, 49], we conduct recognition tasks for both 1-shot and 5-shot, 5-way and 20-way classification on COAD.

The statistical results are visualized in Fig. 9. It is obvious that using more examples for C-way recognition leads to higher accuracy, and 5-way classification is easier than 20-way. The results show that the TML model achieves 51.55% average accuracy in 20-way 1-shot task and even 80.83% in 5-way 1-shot setting, presenting a compelling ability to learn new concepts just in one-shot. The reasonable standard deviations attached in the bar graph indicate the stable performance across 10 random experiment episodes.

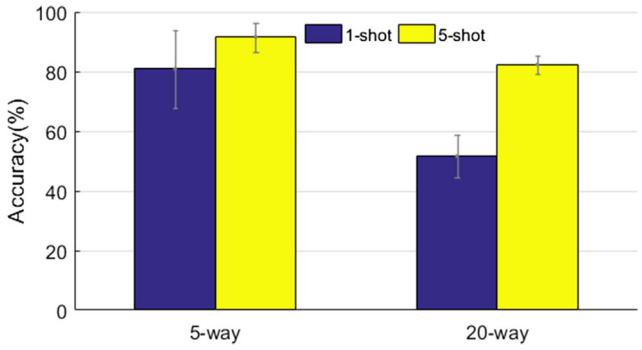


Fig. 9 Few-shot recognition accuracy on COAD

5 Further analysis and discussion

5.1 Parameter investigation

In the proposed TML method, stroke primitives encoding and spatial relationship modeling are two fundamental modules to represent a hand-drawn sketch. For some sketches with rigid spatial relationship between contiguous strokes like COAD symbols, the distance threshold is crucial when modeling their relationships. To explore the influence of distance threshold on recognition accuracy and select an optimal value for experiments, we investigate the issue by a grid search over $\{0, 0.5, 1.0, \cdots, 5.0\}$ on the COAD dataset. The experiments are implemented for 10 episodes and 1100 randomly generated test sketches in 20-way are evaluated in each episode. The evaluation results are gathered in Fig. 10.

It could be found that selecting this threshold carefully is important. Setting it to 0 will make the relationships almost regarded as 'No connection' and result in low accuracy. In contrast, setting it to a large value like 5.0 (the size of sketch drawing window is 10*10) will make the relationships ambiguous and decrease the average accuracy from 82.31% (when threshold is set as 1.0) to 66.14%. According to the statistical results, the relation distance threshold is set to 1.0 in our experiments.

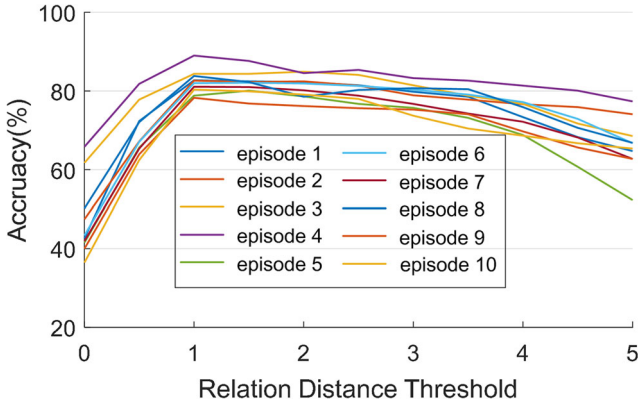


Fig. 10 The recognition accuracy versus varying distance threshold on COAD

5.2 Search strategy modification

For datasets with great similarity across categories such as Omniglot, some different symbols may share the same stroke attributes and spatial relationship sequences, as shown in Fig. 11, which will confuse the machine whenever any of the symbols is retrieved. Fortunately, it is able to memorize all these similar symbol templates and then offer the multiple probable results to users. As shown in Fig. 12, this adaptive search strategy not only provides a sharp improvement in recognition performance, but will play a significant role in practice to remind users of alternative solutions to a particular problem.

5.3 Why does teach machine to learn work?

The cognitive neuroscience foundation The research of artificial intelligence and cognitive neuroscience has been intertwined for long [50]. Derived from the neural networks in computational neuroscience, revolutionary breakthroughs in AI have been achieved in recent years. In contrast to the catastrophic forgetting suffered in neural networks, it should be able to learn novel concepts over multiple timescales and master new tasks without forgetting prior knowledge, which is known as continual learning. Recent perspectives emphasize some concepts related to physical world, such as space, number and objects, which are compatible for inference and prediction by constructing compositional models. More intuitively, human are advantageous than machines in terms of learning new concepts from several examples and leveraging prior knowledge to further inference. These cognitive neuroscience clues lay a foundation for guiding principles of our TML model.

Relationship to existing models Previous works about sketch recognition solely utilize either temporal stroke patterns [31] or spatial image information [29] to analyze the sketch structure. They are almost based on handcrafted stroke geometrical characteristics or image feature descriptors which are lack of generalization in feature embedding. Another important trend is deep models that focus on learning a deep embedding without the combination and inference of the features. Integrating the advantages of pattern encoding and probabilistic inference, our TML model could learn a generalized embedding as well as an inference mechanism. Moreover, it is executed online in real time for lifelong learning or even one-shot learning.

Model generalization Just like the compositional hand-drawn sketches, many artifacts are complex concepts composed of parts. For example, bicycles, cars and scooters share some parts like handlebars and wheels. By combining

Fig. 11 Some symbols of different categories look similar and may share the same stroke sequences. Three sets of similar Omniglot symbols are listed in rows and each includes three categories with 20 samples per class

these parts in novel ways, new artifacts like the segway could be generated [6]. In addition, some other symbols such as spoken words, gestures and sign language also share similarities with our simple visual concepts. Specifically, a spoken word is a sequence of phonemes just like a character is a sequence of strokes. We are eager to see compositional models explaining rapid learning in these potential domains.

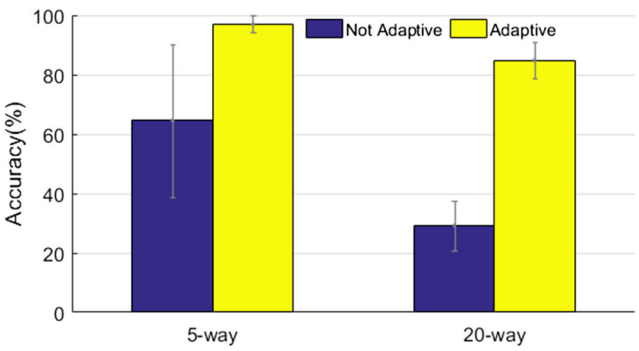


Fig. 12 The performance improvement with adaptive search strategy on Omniglot dataset

6 Conclusion

In this paper, we proposed Teach Machine to Learn, a sketch recognition method with high data efficiency and lifelong learning capacity. Our work suggests that incorporating knowledge representation and probabilistic inference can lead to robust, generalized machine learning models. A further direction would be developing a complete sketch-based human-computer interface in practical applications, using our method as the recognition engine. To improve the recognition performance, spatial relationships could be analyzed in more detail for complicated stroke combinations. Another way is to modify the probabilistic matching

between two symbols that different-length sequences could be matched in a robust way.

Instead of finetuning a mature model for extreme performance on some benchmark tasks, this work intends to investigate and throw light on human few-shot learning, recognition and inference abilities that are yet to be captured by current machine learning models. Our method emphasizes aspects like data efficiency, compositionality, learning to learn and knowledge representation that may be important and promising in the path towards artificial general intelligence and even humanoid intelligence.

Acknowledgments This work was supported by the equipment pre-research sharing technology project of China (No. 41412030301).

References

1. Silver D, Huang A, Maddison CJ et al (2016) Mastering the game of Go with deep neural networks and tree search[J]. Nature 529(7587):484–489
2. Krizhevsky A, Sutskever I, Hinton GE et al (2012) ImageNet classification with deep convolutional neural networks[J]. Neural Inform Process Syst 141(5):1097–1105
3. Nguyen A, Yosinski J, Clune J et al (2015) Deep neural networks are easily fooled: High confidence predictions for unrecognizable images[J]. Comput Vis Pattern Recogn, 427–436
4. Su J, Vargas DV, Sakurai K et al (2019) One pixel attack for fooling deep neural networks[J]. IEEE Trans Evol Comput, 1–1
5. Tirkaz C, Yanikoglu B, Sezgin TM et al (2012) Sketched symbol recognition with auto-completion[J]. Pattern Recogn 45(11):3926–3937
6. Lake BM, Salakhutdinov R, Tenenbaum JB et al (2015) Human-level concept learning through probabilistic program induction[J]. Science 350(6266):1332–1338
7. Vinyals O, Blundell C, Lillicrap TP et al (2016) Matching networks for one shot learning[J]. Neural Inform Process Syst, 3637–3645
8. Aljundi R, Chakravarty P, Tuytelaars T et al (2017) Expert gate: lifelong learning with a network of experts[J]. Comput Vis Pattern Recogn, 7120–7129
9. Ruvolo P, Eaton E (2013) ELLA: an efficient lifelong learning algorithm[C]. In: International conference on machine learning, pp 507–515
10. George D, Lehrach W, Kansky K et al (2017) A generative vision model that trains with high data efficiency and breaks text-based CAPTCHAs[J]. Science 358:6368
11. Holzinger A, Kickmeier-Rust M, Müller H (2019) KANDINSKY patterns as IQ-test for machine learning. In: International cross-domain conference for machine learning and knowledge extraction, lecture notes in computer science LNCS 11713. Springer, Canterbury, pp 1–14, https://doi.org/10.1007/978-3-030-29726-8_1
12. Lecun Y, Bengio Y, Hinton GE et al (2015) Deep learning[J]. Nature 521(7553):436–444
13. Olsen L, Samavati F, Sousa MC et al (2009) Sketch-based modeling: a survey[J]. Comput Graph 33(1):85–103
14. Eitz M, Hildebrand K, Boubekeur T et al (2011) Sketch-based image retrieval: benchmark and bag-of-features descriptors[J]. IEEE Trans Vis Comput Graph 17(11):1624–1636
15. Hu R, Collomosse J (2013) A performance evaluation of gradient field HOG descriptor for sketch based image retrieval[J]. Comput Vis Image Underst 117(7):790–806
16. Forbus KD, Usher J, Chapman V et al (2003) Sketching for military courses of action diagrams[C]. Intelligent User Interfaces, 61–68. https://doi.org/10.1145/604045.604059
17. Hammond T, Logsdon D, Paulson B et al (2010) A sketch recognition system for recognizing free-hand course of action diagrams[C]. Innovative Applications of Artificial Intelligence
18. Paulson B, Hammond T (2008) PaleoSketch: accurate primitive sketch recognition and beautification[C]. Intelligent User Interfaces, 1–10
19. Fonseca MJ, Jorge JA (2000) Using fuzzy logic to recognize geometric shapes interactively[C]. In: IEEE International conference on fuzzy systems, pp 291–296
20. Sezgin TM, Stahovich TF, Davis R et al (2006) Sketch based interfaces: early processing for sketch understanding[C]. In: International conference on computer graphics and interactive techniques
21. Harding PR, Ellis T (2004) Recognizing hand gesture using Fourier descriptors[C]. In: International conference on pattern recognition, pp 286–289
22. Dalal N, Triggs B (2005) Histograms of oriented gradients for human detection[C]. Comput Vis Pattern Recogn, 886–893
23. Ouyang TY, Davis R (2009) A visual approach to sketched symbol recognition[C]. In: International joint conference on artificial intelligence, pp 1463–1468
24. Shechtman E, Irani M (2007) Matching Local Self-Similarities across images and videos[C]. Comput Vis Pattern Recogn, 1–8
25. Oltmans M (2007) Envisioning sketch recognition: a local feature based approach to recognizing informal sketches. Doctoral Dissertation, OAI: oai:dspace.mit.edu:1721.1/40318
26. Rosa MD (2014) New methods, techniques and applications for sketch recognition. Doctoral Dissertation, https://doi.org/10.14273/unisa-304
27. Schneider RG, Tuytelaars T (2014) Sketch classification and classification-driven analysis using Fisher vectors[J]. Int Conf Comput Graph Interact Techn, 33(6)
28. Tümen S, Acer ME, Sezgin TM (2010) Feature extraction and classifier combination for image-based sketch recognition[C]. Sketch Based Interfaces and Modeling, 63–70. https://doi.org/10.2312/SBM/SBM10/063-070
29. Li Y, Hospedales TM, Song Y et al (2015) Free-hand sketch recognition by multi-kernel feature learning[J]. Comput Vis Image Underst, (137), 1–11
30. Ouyang TY (2012) Understanding freehand diagrams: combining appearance and context for multi-domain sketch recognition, Doctoral Dissertation
31. Sezgin TM, Davis R (2007) Sketch interpretation using multiscale models of temporal patterns[J]. IEEE Comput Graph Appl 27(1):28–37
32. Sezgin TM, Davis R (2005) HMM-based efficient sketch recognition[C]. Intell User Interfaces, 281–283
33. Ha D, Eck D (2018) A neural representation of sketch drawings[J]. International Conference on Learning Representations
34. Eitz M, Hays J, Alexa M et al (2012) How do humans sketch objects[J]. Int Conf Comput Graph Interact Techniques, 31(4)
35. Zou C, Yu Q, Du R et al (2018) SketchyScene: richly-annotated scene sketches[C]. Europ Conf Comput Vis, pp 438–454
36. Yu Q, Yang Y, Liu F et al (2017) Sketch-a-net: a deep neural network that beats humans[J]. Int J Comput Vis 122(3):411–425
37. Li Y, Bu R, Sun M et al (2018) PointCNN: convolution on x-transformed points[C]. Neural Inform Process Syst, 820–830
38. Sun Z, Wang C, Zhang L et al (2012) Free hand-drawn sketch segmentation[C]. European Conf Comput Vis, 626–639
39. Zhang J, Chen Y, Li L et al (2018) Context-based sketch classification[C]. Non Photorealistic Animation and Rendering

40. Hu C, Li D, Song Y et al (2018) Sketch-a-classifier: sketch-based photo classifier generation[J]. arXiv: Computer Vision and Pattern Recognition
41. Verma VK, Mishra A, Mishra AK et al (2019) Generative model for zero-shot sketch-based image retrieval[J]. arXiv: Computer Vision and Pattern Recognition
42. Song J, Pang K, Song Y et al (2018) Learning to sketch with shortcut cycle consistency[C]. Comput Vis Pattern Recogn, 801–810
43. Xu P, Huang Y, Yuan T et al (2018) SketchMate: deep hashing for million-scale human sketch retrieval[C]. In: IEEE/CVF Conference on computer vision and pattern recognition, pp 8090–8098. https://doi.org/10.1109/CVPR.2018.00844
44. Lake BM, Salakhutdinov R, Gross J et al (2011) One shot learning of simple visual concepts[J]. Cognit Sci 33:33
45. Girshick R, Donahue J, Darrell T et al (2014) Rich feature hierarchies for accurate object detection and semantic segmentation[J]. Comput Vis Pattern Recogn, 580–587
46. Redmon J, Divvala SK, Girshick R et al (2016) You only look once: unified, real-time object detection[C]. Comput Vis Pattern Recogn, 779–788
47. Liu W, Anguelov D, Erhan D et al (2016) SSD: single shot multibox detector[J]. European Conf Comput Vis, 21–37
48. Lin T, Dollar P, Girshick R et al (2017) Feature pyramid networks for object detection[C]. Comput Vis Pattern Recogn, 936–944
49. Sung F, Yang Y, Zhang L et al (2018) Learning to compare: relation network for few-shot learning[J]. Comput Vis Pattern Recogn, 1199–1208
50. Hassabis D, Kumaran D, Summerfield C et al (2017) Neuroscience-inspired artificial intelligence[J]. Neuron 95(2): 245–258

Chongyu Pan received the B.S. and M.S. degree from the College of Mechatronic Engineering and Automation, National University of Defense Technology, Changsha, China, in 2014 and 2016, where he is currently pursuing the Ph.D. degree. His current research interests include few-shot learning, multi- and cross-modal learning.

Jian Huang received the Ph.D. degrees in the College of Mechatronics and Automation, National University of Defense Technology, Changsha, China, in 2000. She is currently a professor in the College of Intelligence Science. Her current research interests include mission planning, task assignment and distributed simulation.

Jianxing Gong received his Ph.D. degree from National University of Defense Technology, China, in 2007. He is now an associate professor in College of Intelligence Science. His research interests include task planning and simulation.

Cheng Chen received his M.S. degree from the College of Intelligence Science, National University of Defense Technology in 2017. His current research interest is computer vision.