Sketch Recognition with Deep Visual-Sequential Fusion Model

Jun-Yan He[1], Xiao Wu[1#], Yu-Gang Jiang[2], Bo Zhao[1] and Qiang Peng[1]
[1]Southwest Jiaotong University, Chengdu, China
[2]Fudan University, Shanghai, China
{junyanhe1989,zhaobo.cs}@gmail.com,{wuxiaohk,qpeng}@home.swjtu.edu.cn,ygj@fudan.edu.cn

ABSTRACT

In this paper, a deep end-to-end network for sketch recognition, named Deep Visual-Sequential Fusion model (DVSF) is proposed to model the visual and sequential patterns of the strokes. To capture the intermediate states of sketches, a three-way representation learner is first utilized to extract the visual features. These deep features are simultaneously fed into the visual and sequential networks to capture spatial and temporal properties, respectively. More specifically, visual networks are novelly proposed to learn the stroke patterns by stacking the Residual Fully-Connected (R-FC) layers, which integrate ReLU and Tanh activation functions to achieve the sparsity and generalization ability. To learn the patterns of stroke order, sequential networks are constructed by Residual Long Short-Term Memory (R-LSTM) units, which optimize the network architecture by skip connection. Finally, the visual and sequential representations of the sketches are seamlessly integrated with a fusion layer to obtain the final results. Experiments conducted on the benchmark sketch dataset TU-Berlin demonstrate the effectiveness of the proposed method, which outperforms the state-of-the-art approaches.

KEYWORDS

Sketch Recognition, Deep Learning, Residual Learning, Long Short-Term Memory.

1 INTRODUCTION

With the widespread use of smart phones and touch screen devices, it becomes pretty convenient for users to draw sketches on the screen, simply using their fingers. Sketch expresses the general contour information of objects in a straightforward way, instead of struggling to describe it verbally. It has been successfully used for color image synthesis [4, 8], cross-domain image retrieval and recognition [3, 14, 25, 26], and so on. A common and attractive scenario is illustrated in Fig. 1. A user draws the sketch of a pigeon on a touch screen device step by step, and then the sketch together with its corresponding drawing sequence are sent to the cloud platform, which recognizes the object, retrieves similar real images and finally returns them to the user.

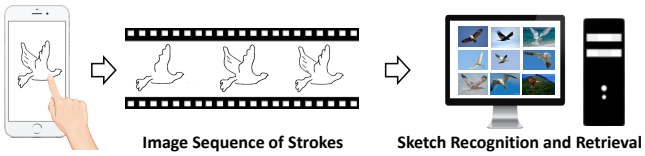


Figure 1: The sketch drew by users is first recognized and then corresponding real objects are retrieved.

However, sketch recognition is a challenging task due to the following limitations. First, unlike traditional real images, sketches are abstract representations of real objects, in which many details of original objects are absent, so that different objects can be visually similar when they are represented by sketches. For example, a standing bird would be depicted as a cock. Second, there exist huge variations between real images and sketches, even they describe the same thing. Since sketch is a free-hand drawing, different users will draw inconsistent sketches for the same object, which can be drawn with various levels of details/abstractions. Third, a sketch only consists of simple lines, curves or dots, without the vivid information such as color and texture, which makes the recognition a difficult task.

Existing sketch recognition methods usually treat the sketches as real images. Global and local features of objects are used to identify their categories. Unfortunately, due to the lack of color and texture information, these approaches achieve unsatisfactory performance. Therefore, the sketch based applications have not been widely applied in reality. Recently, with the prevalence of deep learning, latest studies based on deep features (e.g., [31, 34]) have achieved exciting performance on sketch recognition.

Sketch has two prominent properties: visual pattern and sequential pattern, which are potentially beneficial for sketch recognition. First, the sketch image is composed of a series of strokes. These strokes have certain visual patterns to form the main shape and special details belonging to a specific object, such as the webbed feet of ducks, stripe patterns of zebras, and so on. These stroke visual patterns are crucial for sketch recognition. Second, when drawing a sketch step by step, the strokes follow certain sequential patterns, such as from left to right, top to down, or outside to inside. For example, people usually draw the head of the bird first and then the body and legs. How to utilize these two useful patterns to improve the performance of sketch recognition is the main goal of this paper.

In this paper, we propose an end-to-end deep learning network called Deep Visual-Sequential Fusion model (DVSF) for sketch recognition, which captures both visual and sequential patterns of strokes to boost the performance. The framework of the proposed DVSF model is illustrated in Fig. 2. It mainly consists of four components: representation learner, visual networks, sequential networks and fusion layer. Analogous to a video, a sketch can be treated as

# indicates the corresponding author.
Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
MM'17, October 23–27, 2017, Mountain View, CA, USA.
© 2017 ACM. ISBN 978-1-4503-4906-2/17/10…$15.00
DOI: http://dx.doi.org/10.1145/3123266.3123321

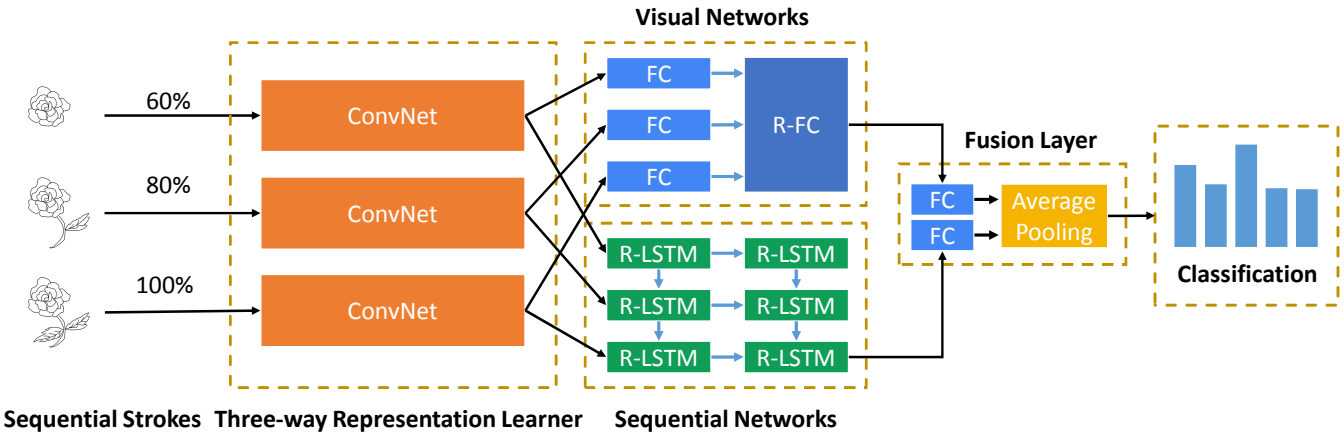


Figure 2: The architecture of Deep Visual-Sequential Fusion model (DVSF). Three-way CNNs are utilized to extract the visual features, which are fed into visual and sequential modeling modules, respectively. Visual networks are novelly proposed by stacking the Residual Fully-Connected (R-FC) layer, while sequential networks are constructed with Residual Long Short-Term Memory (R-LSTM) units. The visual and sequential representations are integrated into the fusion layer to obtain the final prediction.

a sequence of keyframes according to the appearance order of the strokes. To capture the progressive states of sketches, a three-way representation learner is utilized to extract the visual features of sketches. Due to the impressive performance and the flexibility of network configuration, residual networks are adopted in each branch. These deep features induced from representation learner are then fed into visual and sequential networks, which capture visual patterns and sequential patterns, respectively. Visual networks are novelly proposed by stacking the Residual Fully-Connected (R-FC) layer, which integrates ReLU and tanh activation functions to achieve the sparsity and generalization ability. To learn the temporal patterns, sequential networks are constructed by using Residual Long Short-Term Memory (R-LSTM) [18] units, which optimize the network architecture by skip connection. Finally, the representations learned by visual networks and sequential networks are seamlessly integrated with a fusion layer, which consists of two fully-connected layers as classifiers, performing the final score fusion. The performance of the proposed DVSF model is evaluated on the sketch benchmark dataset TU-Berlin, which outperforms state-of-the-art methods. In addition, sketch-based image retrieval is also conducted to further validate the effectiveness of the proposed model. The contributions of this work are summarized as follows:

- An end-to-end network, Deep Visual-Sequential Fusion model (DVSF), is proposed for sketch recognition, which integrates the visual appearance of sketches and the sequential patterns of strokes.
- Visual networks are novelly proposed to integrate the detail information of sketches, by stacking the Residual Fully-Connected (R-FC) layer. It combines the ReLU and tanh activation functions to achieve the sparsity and generalization ability, which improve the discriminative capability of DVSF model.
- To learn the temporal patterns, sequential networks are constructed by using Residual Long Short-Term Memory (R-LSTM) units, which optimize the network architecture by skip connection. To the best of our knowledge, this is the first time that R-LSTM is integrated into the sequential networks to model the temporal patterns of stroke orders.
- Experiments conducted on the sketch benchmark dataset TU-Berlin demonstrate that the proposed method outperforms the state-of-the-art approaches.

The rest of the paper is organized as follows. Related work is reviewed in Section 2. The proposed deep visual-sequential fusion model is elaborated in Section 3. Experimental evaluation and analysis are presented in Section 4. Finally, we conclude this work in Section 5.

2 RELATED WORK

2.1 Sketch Recognition

Sketch recognition has attracted increasing attention of researchers for its broad application prospect. However, it is a challenging task to collect various hand-free sketches until TU-Berlin [6] has been built. Hand-crafted features, encoding methods and classifiers construct the classic technical frameworks for many traditional sketch recognition approaches. In [6], a novel feature is proposed to represent a sketch as a large number of local features, which encode the local orientation estimates. A framework based on dense SIFT features and Fisher vectors is presented for sketch classification [21], which significantly outperforms existing techniques. Because of the huge gap between real images and sketches, general descriptors are not suitable for sketch recognition. To describe a sketch image more effectively, a new descriptor, namely Symmetric-aware Flip Invariant Sketch Histogram (SYM-FISH) [2] is proposed to refine the shape context feature, which achieves much better performance. In [35], a new inter-modality face recognition approach is presented, by reducing the modality gap between features extracted from photos and sketches.

Sketch retrieval is a critical application based on sketch recognition. A representation scheme is proposed in [15] to facilitate

efficient sketch retrieval, which takes into account sketch strokes and local features. Retrieving 3D models from 2D sketches has important applications in computer graphics, information retrieval, and computer vision. A deep learning based approach is proposed in [27] to model different views of 3D objects and 2D sketches, in which two Siamese CNNs are used to learn the representations of different types of inputs: one for views and another for sketches. To bridge the appearance gap between sketches and real images, a framework is proposed in [27], which consists of a new line segment-based descriptor and a new noise impact reduction algorithm. A novel modality-invariant face descriptor is proposed in [11] to retrieve face photos based on a probe sketch.

Recently, deep learning has achieved great success in many computer vision tasks, such as image classification, object detection, and so on. Deep learning framework has also been introduced into the sketch recognition task. In [34], a novel approach is presented to learn the shared latent structures between sketches and real images by using a triplet based network. Another deep learning based sketch recognition framework is proposed in [31], which presents a multi-scale multi-channel structure to encode the sequential ordering in the sketching process. Due to its excellent performance, it is further applied to fine-grain shoe retrieval [30].

2.2 Sequential Modeling

Because there exists certain temporal order when the sketch of an object is drawn, the stroke order pattern is a critical clue for sketch recognition. Recurrent neural networks (RNNs) model the temporal information by connecting previous information to current state. However, there is a long term dependency problem that cannot be addressed by original RNN. A modify RNN, Long Short-Term Memory (LSTM) [10] is proposed to avoid the problem of long term dependency, which has been widely used for activity recognition [16], video classification [5, 23, 28, 32], image caption [19], visual question answering [1, 7, 20, 29], and so on. Meanwhile, the architectural novelty of LSTM includes two dimensional recurrent layers and an effective use of residual connections in deep recurrent networks, offering fast training and multi-layer stacking capability [18].

The most related work is [31], which is a multi-scale multi-channel deep neural network framework for sketch recognition. Unfortunately, because the input batch is not generated in the sketching order, the pattern of stroke order is only weakly captured. Although there are many prior works on sequential modeling, few studies focus on the modeling of stroke order pattern, which motivates this work.

3 DEEP VISUAL-SEQUENTIAL FUSION MODEL

In this section, we will elaborate the four components in DVSF and introduce the training strategy of our model.

3.1 Three-way Representation Learner

Sketch is the abstract representations of real objects, which is drawn stroke by stroke with free hand, just like the handwriting. The process of drawing can be regarded as a continuous stroke sequence with different degrees of stroke completeness. Enlightened by the idea that a video can be represented as a sequence of keyframes after shot boundary detection and keyframe selection, a sketch can be treated as a sequence of stroke images, representing the intermediate status of the sketch. The process of drawing a sketch can be seen as the accumulation of strokes. Therefore, the pattern of the stroke sequence is highly related to the object itself.

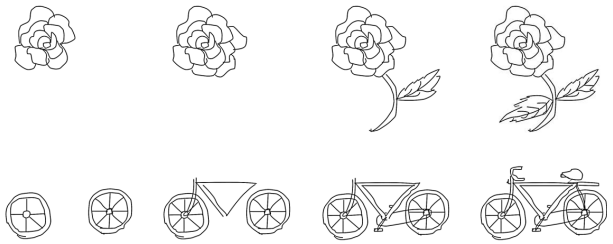


Figure 3: From left to right, the sketch images contain the portions of strokes from 40% to 100%, respectively, with the interval of 20%.

Two examples of stroke sequences from 40% to 100% with the interval of 20% are illustrated in Fig. 3. At the very beginning of sketching, only limited strokes are available. It is difficult to recognize the object from an incomplete sketch, since it only contains small portion of the strokes, that is, part of the object. As can be seen from Fig. 3, the objects are not easy to be recognized when 40% of the strokes are drawn. But when the outline or a complete component of an object has been drawn, i.e., at least half of the strokes is completed, the objects can be roughly estimated. With proper temporal interval, the transition of stroke changes can be well captured.

To balance the temporal changes and speed efficiency, three representative stroke images (i.e., 60%, 80%, 100% of strokes) are selected in this paper. A three-way representation learner is then adopted to capture the temporal sequence of sketches. The 18-layer residual networks (Resnet-18) [9] are deployed to learn the representation of stroke images with different degrees of completeness. The extracted deep features represent the intermediate status of the object, which will be used for further processing.

3.2 Visual Networks

Once the deep convolutional features of the stroke images are obtained, visual networks are adopted to learn the visual appearance patterns of the sketches, as illustrated in Fig. 2. More specifically, there are three Fully-Connected (FC) layers followed by a novel Residual Fully-Connected Layer (R-FC) in visual networks. This structure can enhance the discrimination of features by fusing the partial and complete sketch features, which is beneficial for the classification.

The structure of Residual Fully-Connected Layer is illustrated in Fig. 4. Deep features are activated by two activation functions, ReLU and tanh, to map the features to different non-linear spaces, so that the features can be well discriminated. ReLU function sets negative neurons to zero, offering the sparsity and reducing parameter dependencies to avoid the problem of over-fitting. Meanwhile, tanh is a zero mean function which maps the input value to $[-1, 1]$. The fixed range brings the stability in the training stage, and it can generate different non-linear spaces compared to ReLU function. The

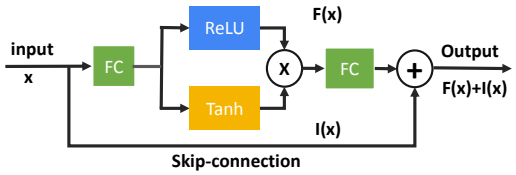


Figure 4: The architecture of Residual Fully-Connected (R-FC) layer.

most straightforward way to combine these two activation outputs is with addition operation. However, tanh function is non-linear and it easily activates the neurons to saturation state. The gradient of tanh function changes slowly in the saturation regime, leading to the problem of gradient vanishing. In addition, tanh suppresses the sparsity induced from ReLU, which is critical for removing the noise and boosting the performance. We have conducted several experiments to explore appropriate fusion methods. The results show that fusion by addition operation easily leads to the loss explosion, even though the learning rate is set to a very small value (e.g., 10^{-3}). Finally, element-wise multiplication operation is selected to integrate the activation outputs.

To accelerate the training speed and boost the performance, residual architecture is adopted in our R-FC unit. An extra skip connection is adopted to convert the optimization objective from the desired underlying mapping to a residual mapping. An identity mapping is added to this unit, so that the signal can be propagated from one unit to the next layer by skip connection.

R-FC is formulated as:

$$\begin{aligned} l &= F(x), \\ Y &= F(\mathrm{ReLU}(l) \cdot \tanh(l)) + I(x), \end{aligned} \tag{1}$$

where x is the input, l and Y are the immediate output and final output of R-FC. $F(x)$ donates the fully-connected layer and $I(x)$ refers to the identity mapping.

3.3 Sequential Networks

To model the temporal patterns of the sketches, sequential networks are proposed, which consist of several novelly designed Residual Long Short-Term Memory (R-LSTM) [18] units. Similar to the original LSTM units, R-LSTM also contains three gates (i.e., in, out and forget gates) to control the flow of information into or out of their memories. These gates are implemented with the logistic function (e.g., sigmoid) to compute a value within the range of 0 to 1. However, the sigmoid function in original LSTM will easily activate the neurons to saturation state, leading to gradient vanishing, even though the forget gate allows the gradient to stay stable. To prevent this problem, we add the ReLU mapping as an additional data flow transmission channel in R-LSTM, so that the data flow can skip LSTM units in the training stage. The ReLU mapping not only offers the additional transmission channel but also produces the sparsity of the LSTM networks to improve the generalization ability and accuracy. Finally, the structure of sequential networks is also optimized with a skip connection, which boosts the overall performance. In addition, to better capture the hierarchical structure of a sequence, multiple layers of R-LSTM are constructed in our sequential networks, as shown in Fig. 2.

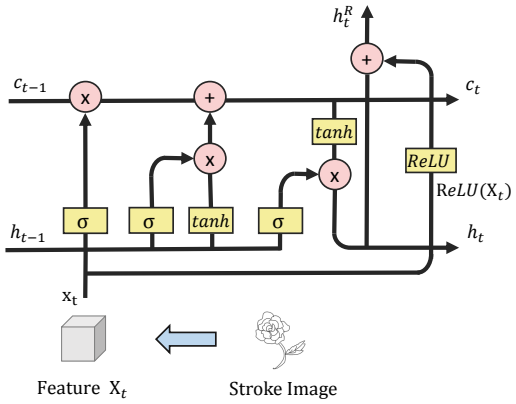


Figure 5: The architecture of Residual LSTM (R-LSTM) unit with ReLU mapping.

The architecture of R-LSTM is illustrated in Fig. 5, which is formulated as follows:

$$\begin{aligned} \begin{pmatrix} \mathbf{i}_t \\ \mathbf{f}_t \\ \mathbf{o}_t \\ \mathbf{g}_t \end{pmatrix} &= \begin{pmatrix} \sigma \\ \sigma \\ \sigma \\ \tanh \end{pmatrix} M \begin{pmatrix} \mathbf{h}_{t-1} \\ \mathbf{x}_t \end{pmatrix}, \\ \mathbf{c}_t &= \mathbf{f}_t \odot \mathbf{c}_{t-1} + \mathbf{i}_t \odot \mathbf{g}_t, \\ \mathbf{h}_t &= \mathbf{o}_t \odot \tanh(\mathbf{c}_t), \\ \mathbf{h}_t^R &= \mathbf{h}_t + \mathrm{ReLU}(\mathbf{x}_t), \end{aligned} \tag{2}$$

where x_t is the features generated by the last convolutional layer of CNN at time step t when drawing a sketch. i_t, o_t and f_t are the input, output and forget gates, respectively, and g_t is the memory cell. c_t and h_t encode the cell state and hidden state, respectively. σ is the sigmoid activation function $\sigma(x) = 1/(1 + \exp -x)$ and $\odot$ presents the element-wise multiplication with a gate value. $tanh$ donates the hyperbolic tangent, and the ReLU mapping is adopted to compute h_t^R. It is worthy to note that h_t is transmited through the same level of R-LSTM at different time steps, and h_t^R is passed to the next level of R-LSTM. R-LSTM involves a transformation $M : R^a \rightarrow R^b$, which consists of $a \times b$ trainable parameters with $a = d + D$ and $b = 4d$, where d is the dimension of i_t, o_t, f_t, g_t, c_t and h_t, and D denotes the dimension of the input features.

Since the outputs of sequential networks at intermediate time steps contain both temporal and incomplete spatial information of sketches, it will confuse the classifier and lead to performance drop. Therefore, the output of the second layer of sequential networks at the last time step is used for classification, as illustrated in Fig. 2.

3.4 Fusion Layer

Since visual networks and sequential networks model the stroke patterns from different aspects, the produced representations complement each other. For better classification performance, a fusion layer is proposed to integrate these two kinds of representations, which employs two FC layers to perform the classification and an average pooling to produce the final result. The fusion layer is defined as follows:

$$\pi(x)_i = \frac{\exp((\Omega_v(x)_i + \Omega_s(x)_i)/2)}{\sum_{i=1}^{K} \exp((\Omega_v(x)_i + \Omega_s(x)_i)/2).} \tag{3}$$

$\Omega_v(x)$ and $\Omega_s(x)$ donate the classification results for visual and sequential networks, respectively. K and i represent the length and the ith dimension of the output vector of FC layer, respectively. The fusion output $\pi(x)$ is used to predict the sketch category by Softmax function. In the training stage, the final scores and the ground truth labels are fed into an objective function to compute the loss. The whole DVSF model is formulated as follows:

$$L = \arg\min \|\pi(x) - y\| + \lambda_1 \|w\|_2 \tag{4}$$

$\|\pi(x) - y\|$ donates the loss with average pooling fusion, and w refers to all parameters involved in DVSF model.

3.5 Model Training

The proposed DVSF model integrates three networks (i.e., representation learner, visual networks and sequential networks) for sketch recognition. Due to the difference among these networks, it is a challenging task to train DVSF model. In this paper, a three-stage training strategy is adopted to train it.

First, instead of employing pre-train networks for three branches of the three-way representation learner, it is first trained with different portions of strokes to learn more representative features. Three-way representation learner combined with average pooling (denoted as RLA-3) is adopted to train it, which is formulated as:

$$\arg\min \|\frac{1}{n}\sum_{j=1}^{n} \omega_j(x) - y\| + \|w_r\|, \tag{5}$$

where x is the input sketch and y donates the category label of the sketch. $\omega_j(x)$ and w_r denote the classification result of the j-th branch and the weight of the n-way representation learner, respectively. In this paper, n is set to 3 since three-way representation learner is used.

Second, we remove the average pooling fusion layer and the last FC layer of each branch of the pre-trained RLA-3 model, and then combine them with visual networks:

$$\arg\min \|\Omega_v(x) - y\| + \lambda\|w_r\|_2 + \lambda\|w_v\|_2, \tag{6}$$

and sequential networks:

$$\arg\min \|\Omega_s(x) - y\| + \lambda\|w_r\|_2 + \lambda\|w_s\|_2, \tag{7}$$

where w_v and w_s donate the weights of visual networks and sequential networks, respectively. The weight of the three-way representation learner w_r is obtained in the first step training.

Finally, three components of DVSF model are then combined and trained jointly:

$$\arg\min \|\pi(x) - y\| + \lambda\|w_r\|_2 + \lambda\|w_v\|_2 + \lambda\|w_s\|_2. \tag{8}$$

Once the DVSF model is trained, the prediction $\hat{y}$ can be obtained as follows:

$$\hat{y} = \pi(x). \tag{9}$$

3.6 Implementation Details

In sequential networks, three FC layers with 512 hidden units are adopted to capture the features from the three-way representation learner, then a concatenat layer is used to combine the three outputs as one, which is fed into the R-FC layer. Sequential networks are built by stacking two layers of R-LSTM. The number of hidden units of R-LSTM is also set to 512.

Table 1: Effect of stroke completeness

Percentage of strokes	Acc@Top-1	Acc@Top-5
40%	55.3%	81.3%
50%	62.4%	85.9%
60%	65.1%	90.0%
70%	67.4%	90.0%
80%	70.6%	91.7%
90%	73.0%	91.2%
100%	75.1%	93.3%
80%+90%+100%	75.7%	93.6%
40%+60%+80%+100%	73.0%	92.5%
60%+80%+100%	**76.5%**	**94.9%**

(1) The results of the first part are obtained by single Resnet-18 with different percentages of stroke completeness. (2) The results of the second part are obtained by k-way representation layer combined with average pooling.

The DVSF model is implemented with the publicly available machine learning toolkit, Torch[1]. The mini-batch size is set to 32 and the initial learning rate is 0.1, which will be decreased by 10 times every 30 epochs and the total number of training epochs is set to 90. All experiments are trained by stochastic gradient descent (SGD) with 0.9 momentum and 0.0001 weight decay. It takes around 10 hours to train the DVSF model on a workstation with Intel I7 processor and dual NVIDIA TITAN X GPUs.

4 EXPERIMENTS

4.1 Dataset and Evaluation Metric

Dataset. TU-Berlin sketch dataset [6] is a benchmark dataset, which has been widely used to evaluate the performance of sketch recognition [6, 21, 30, 31, 34]. There are 250 object categories in the dataset, which cover the most commonly used daily objects. In each category, 80 sketches are collected by person drawing. Totally, there are 20,000 sketches in the dataset. In the experiments, the sketch dataset is split into three parts for training, testing and validation, containing 70%, 20% and 10% of sketches, respectively.

Evaluation Metric. In our sketch recognition task, the recognition accuracy is used as the evaluation metric. The Acc@K is the percentage of sketches whose true-match photos are ranked in the top K results. Since similar sketches are easily to be incorrectly recognized, top-1 and top-5 accuracy is employed to evaluate the model.

4.2 Effect of Individual Components

In this subsection, we will analyze the performance of individual components in DVSF model.

4.2.1 Effect of Stroke Completeness. To evaluate the effect of the stroke completeness, we conduct experiments by changing the percentage of stroke completeness from 40% to 100%. Residual networks with 18 layers (Resnet-18) [9] are adopted for sketch classification. The results are listed in Table 1. From this table, we can see that the completeness percentage has a substantial impact on the recognition accuracy. The top-1 and top-5 accuracy is only 55.3% and 81.3%, respectively, when 40% of strokes are available. When incomplete strokes or rough contours are partially

[1] http://torch.ch/

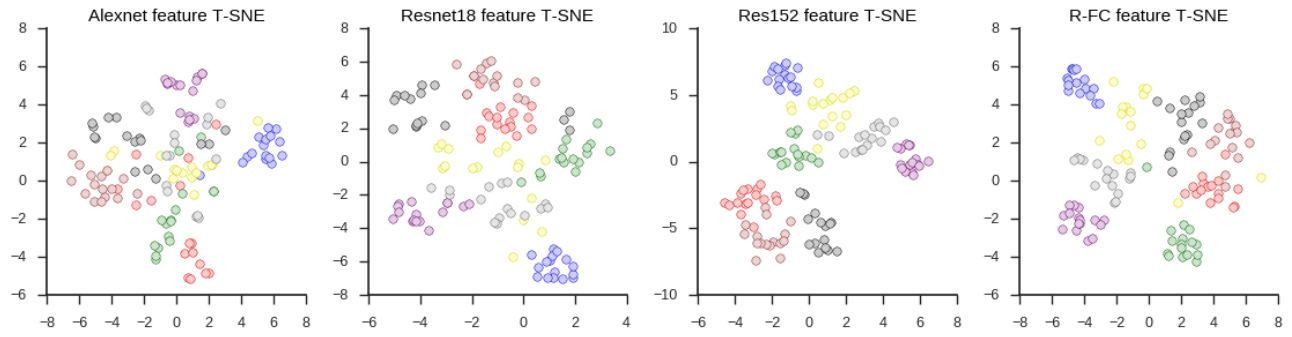


Figure 6: The distribution maps based on T-SNE demonstrate the discrimination of the features.

Table 2: Performance comparison of individual components in DVSF model

Method	Acc@Top-1	Acc@Top-5
Resnet-18+FC	77.1%	94.5%
Resnet-18+R-FC	78.4%	94.4%
Resnet-18+LSTM	76.7%	93.5%
Resnet-18+R-LSTM-I	78.2%	94.5%
Resnet-18+R-LSTM-R	78.7%	94.7%
Resnet-18+R-LSTM-R+R-FC	**79.6%**	**95.3%**

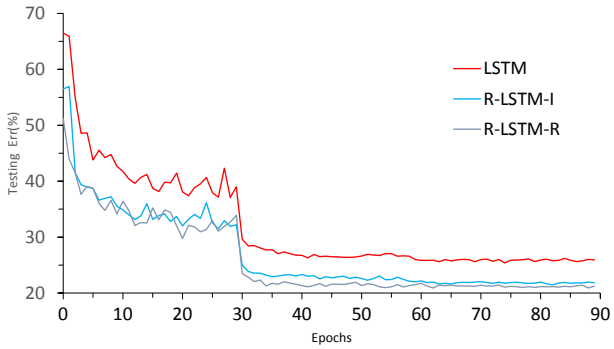


Figure 7: The effect of training epochs.

appeared, the information is pretty limited, from which it is difficult to recognize the objects. The accuracy increases when more strokes are included, especially for the top-1 performance. The top-1 and top-5 accuracy reaches 75.1% and 93.3%, respectively, when all strokes of a sketch are fully appeared. It becomes easier to recognize the objects, when more strokes are included.

In addition, we test the performance using different combinations of completeness percentage of stroke images, which are listed in the second part of Table 1. Three-way representation layer combined with average pooling (RLA-3) is adopted for classification, and two intervals, i.e., 10% and 20% are employed in the experiments. The top-1 accuracy of RLA-3 with the interval of 10% is 75.5%, having little improvement compared to Resnet-18. RLA-3 with the interval of 20% achieves better performance, which has 76.5% and 94.9% for top-1 and top-5 accuracy, respectively. Because the outlines of three sketches are very close with small intervals, and models trained by the sketches obtain similar weights and outputs, the fusion results have limited improvement compared to single Resnet-18. The average pooling fusion integrates scores produced by low completeness percentages of stroke images, leading to the drop of final accuracy of RLA-3. We can find that the accuracy of 40% of strokes is pretty low, so the final fusion score of 4-way representation learner is dropped after integrating with the score produced by 40% of branch. Therefore, we do not add the intermediate stages lower than 60%. According to the experimental results, the interval of 20% is selected to construct the input batch.

4.2.2 Effect of R-FC. First, we evaluate the performance of R-FC, which is listed in Table 2. FC layer is used to fuse the features produced by three-way representation learner, which acts as the baseline (Resnet-18+FC). When FC layer is replaced with the proposed R-FC (Resnet-18+R-FC), it achieves 78.4% top-1 accuracy. Two activations of R-FC map the features to different non-linear spaces, making it more effective than FC in the global and local feature fusion.

To explore the feature discrimination of R-FC, experiments on feature visualization are conducted based on T-SNE [17]. T-SNE reduces the features from high dimensions to two dimensions, so that the feature distribution can be illustrated in a 2D coordinate system, from which the distance between categories can be observed. We randomly select eight categories from the dataset, which are labeled with different colors. The features using AlexNet, Resnet-15, Resnet-152 and Resnet-18+R-FC are extracted, respectively. The results are illustrated in Fig. 6. From this figure, we can see that R-FC features have the best performance. The features are closely grouped into several clusters, and different clusters have relatively obvious boundaries. It is easy to find that Resnet-18+R-FC features are discriminative, which outperform Resnet-18+FC. This demonstrates that R-FC captures the visual patterns of sketches well.

4.2.3 Effect of R-LSTM. Here, we verify the performance of R-LSTM. The performance comparison is also listed in Table 2. Resnet-18+LSTM utilizes LSTM units to build sequential networks and model the stroke order of sketches. It has similar performance as RLA-3. Resnet-18+R-LSTM is similar to Resnet-18+LSTM, but LSTM units are replaced by R-LSTM units. When the Residual LSTM is adopted, the performance is boosted. R-LSTM-I and R-LSTM-R adopt identity mapping and ReLU mapping as the skip connection, respectively. ReLU activation suppresses the negative values and offers the sparsity for the networks. The results show that ReLU mapping is better than identity mapping. In addition, the relationship between training epochs and testing error is illustrated

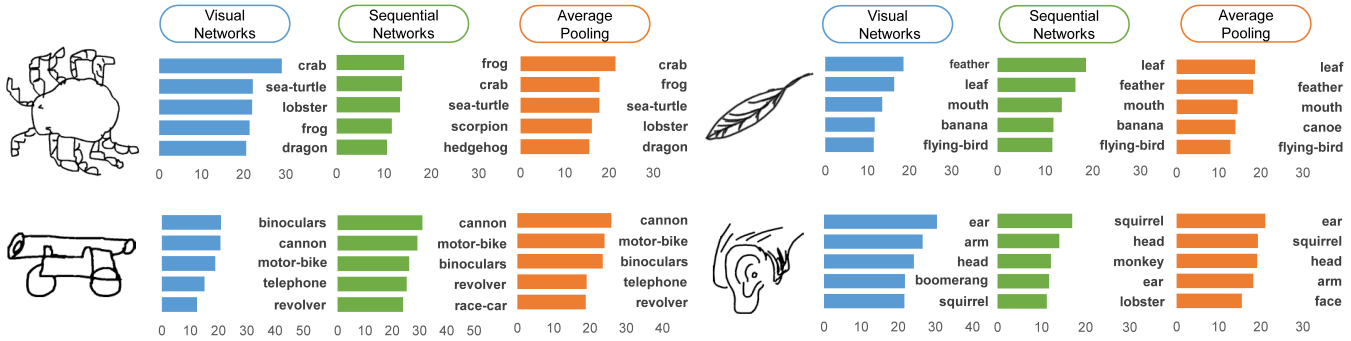


Figure 8: Top-5 scores of visual networks, sequential networks and average pooling, respectively.

in Fig. 7, which demonstrates that LSTM with residual architecture (R-LSTM) can learn faster and better than original LSTM.

4.3 Case Study

In this subsection, we give four examples to demonstrate the top-5 scores of visual networks, sequential networks and average pooling, respectively, which are illustrated in Fig. 8. In the experiments, we utilize R-FC unit and R-LSTM-R units to build visual networks and sequential networks, respectively, and they are used to extract the scores. As we can see, the distributions of top-5 scores produced by visual networks and sequential networks are inconsistent, which complement each other, since they model the information from different aspects. Average pooling fusion is a common way to combine these two factors. For example, the sketch of crab is correctly recognized by visual networks, but falsely detected by sequential networks, which regard it as a frog. On the contrary, the sketch of cannon is falsely recognized as binoculars by visual networks. To some extent, they are visually similar. Fortunately, it is correctly recognized with sequential networks based on the pattern of stroke order. With the combination of visual networks and sequential networks, they can correctly recognize these objects.

4.4 Comparison with CNN Baselines

Since DVSF model is derived from CNN, we compare it with four popular CNN baselines to evaluate its performance: 1) **AlexNet** [13], the first deep networks for computer vision with five convolutional and three fully-connected layers; 2) **VGGNet-BN** VGGNet [22] with extra batch normalization layer; 3) **Resnet-18, Resnet-34 and Resnet-152** [9] are residual networks with 18, 34 and 152 layers, respectively. 4) **WRN28-12** [33] is the 28 layers of Wide Residual Net (WRN) with the width factor as 12. The source code of WRN is originally employed to classify CIFAR-10 and CIFAR-100 datasets [12], in which the image resolution is only $32 * 32$. For images with $224 * 224$ or other sizes, the last average pooling size on the top of WRN is not suitable for previous feature maps. To utilize WRN for sketch recognition, we add three $3 * 3$ max pooling layers after the stages of one, two and three, respectively, to down sample the feature maps. The performance comparison is listed in Table 3.

AlexNet obtains only 67.1% top-1 and 86.7% top-5 accuracy, which has the worst performance among all deep models due to its limited parameters. The small scale of the networks is not able to capture such complex patterns for sketch appearance. VGGNet-BN has more layers and consequently a large number of parameters are involved, which boosts the performance. It achieves 75.6% top-1 and 93.7% top-5 accuracy, more than 10% improvement compared to AlexNet. Resnet-18 has slightly worse performance compared to VGGNet-BN. Meanwhile, Resnet-34 and Resnet-152 have slightly better performance. When the scale of networks is large enough, the contribution of increased layers is relatively small, even though residual networks have 152 layers. It is a solid justification for us to utilize Resnet-18 as the branch of three-way representation learner to trade-off the computational cost and the performance. WRN28-12 achieves pretty good performance, since it is a very large scale deep network. By integrating the visual appearance and temporal patterns, the proposed DVSF model outperforms other baseline methods.

Table 3: Performance comparison with CNN baselines and state-of-the-art approaches.

Method	Acc@Top-1	Acc@Top-5
AlexNet [13]	67.1%	86.7%
VGGNet-BN [22]	75.6%	93.7%
Resnet-18 [9]	75.1%	93.3%
Resnet-34 [9]	76.2%	93.8%
Resnet-152 [9]	76.3%	93.8%
WRN28-12 [33]	76.9%	94.3%
BOF+SVM [6]	54.3%	N/A
FisherVector [21]	66.0%	N/A
SketchANet [31]	74.1%	N/A
SketchANet [30]	77.2%	N/A
DVSF	**79.6%**	**95.3%**

4.5 Comparison with State-of-the-art Methods

To verify the effectiveness of the proposed DVSF model, we compare it with the following state-of-the-art approaches: 1) **SketchANet** [31] is a multi-scale and multi-channel framework for sketch recognition. Two versions of SketchANet are used for comparison, the original one [31] and the pre-trained version [30]. For better performance, SketchANet [30] is pre-trained on a large number of edge images extracted from ImageNet dataset. 2) **BOF+SVM** [6] uses the bag-of-feature model to encode the features and classifies them with SVM. 3) **FisherVector** [21] is based on SIFT descriptor and Fisher Vector is used to encode the local features. Since the compared methods [6, 21, 30, 31] do not report the top-5 accuracy, and the source codes for these methods are not publicly available,



Figure 9: Sketch based image retrieval.

the results are directly imported from their works. We can only compare the top-1 performance in this experiment. The details of the results are listed in Table 3.

Generally, our proposed DVSF model performs the best, which significantly outperforms the CNN based method SketchANet, and the hand-crafted feature based methods, BOF+SVM and FisherVector. BOF+SVM and FisherVector only obtain 54% and 66% top-1 accuracy, respectively, much lower than DVSF model and CNN based methods. Compared to traditional hand-crafted features, deep learning approaches automatically learn the sketch patterns, offering promising performance. The original and pre-trained SketchANet achieves 74.1% and 77.2% for top-1 accuracy, respectively. It demonstrates that the multi-channel and multi-scale structure is robust for sketch recognition. With the assistance of ImageNet dataset, the pre-trained SketchANet has better performance then the original version. SketchNet [34] is another recently proposed sketch classification framework, which achieves excellent performance. Unfortunately, because there is no open source code available, and it is trained with private auxiliary dataset, we cannot compare with it. Our DVSF model achieves the best performance, which outperforms SketchANet even though it is trained without auxiliary dataset.

4.6 Sketch-based Image Retrieval

To further verify the performance of sketch recognition, we build a semantic retrieval system based on DVSF model. A real image dataset is collected from Google image search with 250 categories, corresponding to the same categories in TU-Berlin dataset. Each category contains more than 100 real images, and the whole dataset includes more than 30,000 images. We train a GoogLeNet [24] model using the real image dataset, then extract the score vectors of real images. Each dimension of the score vector is treated as a visual word, and then an inverted index is constructed based on them. In the retrieval stage, we utilize the DVSF model to extract the sketch score vector and use the top-5 scores as a "visual sentence" to retrieve the results.

The retrieval results for six examples are illustrated in Fig. 9. From this figure, we can see that the semantic retrieval offers high precision. The retrieved results of the windmill, TV set and alarm clock are 100% semantically correct. For the sketch of bicycle, although the retrieved results are visually similar to it, some of them are not exactly the bicycle. Cannon is mistakenly recognized as bicycle, because it also has two wheels. Two returned images are motor bicycles, which are visually similar. One interesting result is that two bicycle pictures printed on the ground are also retrieved, which are semantically and visually relevant to the bicycle. The sketch retrieval for standing bird is not so satisfactory. The returned results contain different species of birds, including duck and cock. However, fine-grained object recognition is still a challenging task. For certain categories, it is even hard for human beings to discern the difference. The sketch only contains limited information and critical details are partially or even totally absent, from which it is difficult to distinguish the fine grained categories.

5 CONCLUSIONS

In this paper, a novel sketch recognition model is proposed, which is based on CNN and R-LSTM networks to capture the visual and temporal patterns of the sketches. The experiments demonstrate that the proposed model outperforms the cutting-edge sketch recognition methods. Although promising performance has been achieved, there are still many issues to be further explored, such as the stroke interval, the structure of R-LSTM and the score fusion scheme. In addition, the sketches are carefully drawn in current dataset, which are relatively comprehensive. However, in reality, users usually draw rough sketches with less and inaccurate strokes on touch screen devices. To explore this scenario, we will further refine the framework and design more practical solutions in our future work.

ACKNOWLEDGMENTS

This work was supported in part by the National Natural Science Foundation of China (61373121 and 61622204), Program for Sichuan Provincial Science Fund for Distinguished Young Scholars (No. 13QNJJ0149), and the Sichuan Science and Technology Innovation Seedling Fund (No. 2017RZ0015).

REFERENCES

[1] Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C Lawrence Zitnick, and Devi Parikh. 2015. Vqa: Visual question answering. In *Proc. IEEE Int. Conf. Comput. Vis.* 2425–2433. DOI: https://doi.org/10.1109/ICCV.2015.279

[2] Xiaochun Cao, Hua Zhang, Si Liu, Xiaojie Guo, and Liang Lin. 2013. Sym-fish: A symmetry-aware flip invariant sketch histogram shape descriptor. In *Proc. IEEE Euro. Conf. Comput. Vis.* 313–320. DOI: https://doi.org/10.1109/ICCV.2013.46

[3] Yang Cao, Hai Wang, Changhu Wang, Zhiwei Li, Liqing Zhang, and Lei Zhang. 2010. Mindfinder: interactive sketch-based image search on millions of images. In *in Proc. ACM Conf. Multimedia*. 1605–1608. DOI: https://doi.org/0.1145/1873951.1874299

[4] Tao Chen, Ming-Ming Cheng, Ping Tan, Ariel Shamir, and Shi-Min Hu. 2009. Sketch2Photo: internet image montage. *ACM Trans. Graphics* 28, 5 (2009), 124:1–124:10. DOI: https://doi.org/10.1145/1618452.1618470

[5] Zhi-Qi Cheng, Xiao Wu, Yang Liu, and Xian-Sheng Hua. 2017. Video2Shop: Exact Matching Clothes in Videos to Online Shopping Images. In *Proc. IEEE Conf. on Comput. Vis. and Pattern Recog.*

[6] Mathias Eitz, James Hays, and Marc Alexa. 2012. How do humans sketch objects? *ACM Trans. Graphics* 31, 4 (2012), 44–1. DOI: https://doi.org/10.1145/2185520.2185540

[7] Haoyuan Gao, Junhua Mao, Jie Zhou, Zhiheng Huang, Lei Wang, and Wei Xu. 2015. Are you talking to a machine? dataset and methods for multilingual image question. In *Adv. Neural Inf. Process. Syst. Workshop on Statistical Machine Translation*. 2296–2304.

[8] Yağmur Güçlütürk, Umut Güçlü, Rob van Lier, and Marcel A. J. van Gerven. 2016. Convolutional Sketch Inversion. In *ICCV Workshop on LSMDC*. 810–824.

[9] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep Residual Learning for Image Recognition. In *Proc. IEEE Conf. Comput. Vis. Pattern Recog.* 770–778. DOI: https://doi.org/10.1109/CVPR.2016.90

[10] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation* 9, 8 (1997), 1735–1780. DOI: https://doi.org/10.1162/neco.1997.9.8.1735

[11] Hamed Kiani Galoogahi and Terence Sim. 2012. Photo Retrieval by Sketch Example. In *Proc. ACM. Conf. on Multimedia*. 949–952. DOI: https://doi.org/10.1145/2393347.2396354

[12] Alex Krizhevsky and Geoffrey Hinton. 2009. Learning multiple layers of features from tiny images. *Tech Report* (2009).

[13] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. 2012. ImageNet Classification with Deep Convolutional Neural Networks. In *Proc. Adv. Neural Inf. Process. Syst.* 1097–1105. DOI: https://doi.org/10.1145/3065386

[14] Qingshan Liu, Xiaoou Tang, Hongliang Jin, Hanqing Lu, and Songde Ma. 2005. A nonlinear approach for face sketch synthesis and recognition. In *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, Vol. 1. 1005–1010. DOI: https://doi.org/10.1109/CVPR.2005.39

[15] Chao Ma, Xiaokang Yang, Chongyang Zhang, Xiang Ruan, Ming-Hsuan Yang, and Omron Coporation. 2013. Sketch Retrieval via Dense Stroke Features. In *Proc. British Mach. Vis. Conf.* 64–73. DOI: https://doi.org/10.1016/j.imavis.2015.11.007

[16] Shugao Ma, Leonid Sigal, and Stan Sclaroff. 2016. Learning Activity Progression in LSTMs for Activity Detection and Early Detection. In *Proc. IEEE Conf. Comput. Vis. Pattern Recog.* 1942–1950. DOI: https://doi.org/10.1109/CVPR.2016.214

[17] Laurens van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-SNE. *Journal of Machine Learning Research* 9, Nov (2008), 2579–2605.

[18] Aaron van den Oord, Nal Kalchbrenner, and Koray Kavukcuoglu. 2016. Pixel recurrent neural networks. *arXiv preprint arXiv:1601.06759* (2016).

[19] Pingbo Pan, Zhongwen Xu, Yi Yang, Fei Wu, and Yueting Zhuang. 2016. Hierarchical Recurrent Neural Encoder for Video Representation With Application to Captioning. In *Proc. IEEE Conf. Comput. Vis. Pattern Recog.* 1029–1038. DOI: https://doi.org/10.1109/CVPR.2016.117

[20] Mengye Ren, Ryan Kiros, and Richard Zemel. 2015. Exploring models and data for image question answering. In *Proc. Adv. Neural Inf. Process. Syst.* 2953–2961.

[21] Rosália G Schneider and Tinne Tuytelaars. 2014. Sketch classification and classification-driven analysis using fisher vectors. *ACM Trans. Graphics* 33, 6 (2014), 174. DOI: https://doi.org/10.1145/2661229.2661231

[22] Karen Simonyan and Andrew Zisserman. 2015. Very deep convolutional networks for large-scale image recognition. In *Proc. Int. Conf. Learning Representations*. DOI: https://doi.org/10.1109/ACPR.2015.7486599

[23] Nitish Srivastava, Elman Mansimov, and Ruslan Salakhudinov. 2015. Unsupervised Learning of Video Representations using LSTMs. In *Proc. Int. Conf. on Machine Learning*. 843–852.

[24] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. 2015. Going deeper with convolutions. In *Proc. IEEE Conf. Comput. Vis. Pattern Recog.* 1–9. DOI: https://doi.org/10.1109/CVPR.2015.7298594

[25] Xiaoou Tang and Xiaogang Wang. 2003. Face sketch synthesis and recognition. In *Proc. IEEE Conf. Comput. Vis. Pattern Recog.* 687–694. DOI: https://doi.org/10.1109/ICCV.2003.1238414

[26] Xiaoou Tang and Xiaogang Wang. 2004. Face sketch recognition. *IEEE Trans. Circuits Syst. Video Technol.* 14, 1 (2004), 50–57. DOI: https://doi.org/10.1109/TCSVT.2003.818353

[27] Fang Wang, Le Kang, and Yi Li. 2015. Sketch-based 3d shape retrieval using convolutional neural networks. In *Proc. IEEE Conf. Comput. Vis. Pattern Recog.* 1875–1883. DOI: https://doi.org/10.1109/CVPR.2015.7298797

[28] Zuxuan Wu, Xi Wang, Yu-Gang Jiang, Hao Ye, and Xiangyang Xue. 2015. Modeling spatial-temporal clues in a hybrid deep learning framework for video classification. In *Proc. ACM Conf. Multimedia*. 461–470. DOI: https://doi.org/10.1145/2733373.2806222

[29] Zichao Yang, Xiaodong He, Jianfeng Gao, Li Deng, and Alex Smola. 2016. Stacked attention networks for image question answering. In *Proc. IEEE Conf. Comput. Vis. Pattern Recog.* 21–29. DOI: https://doi.org/10.1109/CVPR.2016.10

[30] Qian Yu, Feng Liu, Yi-Zhe Song, Tao Xiang, Timothy M. Hospedales, and Chen-Change Loy. 2016. Sketch Me That Shoe. In *Proc. IEEE Conf. Comput. Vis. Pattern Recog.* 799–807. DOI: https://doi.org/10.1109/CVPR.2016.93

[31] Qian Yu, Yongxin Yang, Yi-Zhe Song, Tao Xiang, and Timothy M. Hospedales. 2015. Sketch-a-Net that Beats Humans. In *Proc. British Mach. Vis. Conf.* 7.1–7.12. DOI: https://doi.org/10.1007/s11263-016-0932-3

[32] Joe Yue-Hei Ng, Matthew Hausknecht, Sudheendra Vijayanarasimhan, Oriol Vinyals, Rajat Monga, and George Toderici. 2015. Beyond short snippets: Deep networks for video classification. In *Proc. IEEE Conf. Comput. Vis. Pattern Recog.* 4694–4702. DOI: https://doi.org/10.1109/CVPR.2015.7299101

[33] Sergey Zagoruyko and Nikos Komodakis. 2016. Introduction to Wide Residual Networks. In *Proc. British Mach. Vis. Conf.* DOI: https://doi.org/10.13140/RG.2.2.18632.72962

[34] Hua Zhang, Si Liu, Changqing Zhang, Wenqi Ren, Rui Wang, and Xiaochun Cao. 2016. SketchNet: sketch classification with web images. In *Proc. IEEE Conf. Comput. Vis. Pattern Recog.* 1105–1113. DOI: https://doi.org/10.1109/CVPR.2016.125

[35] Wei Zhang, Xiaogang Wang, and Xiaoou Tang. 2011. Coupled information-theoretic encoding for face photo-sketch recognition. In *Proc. IEEE Conf. Comput. Vis. Pattern Recog.* 513–520. DOI: https://doi.org/10.1109/CVPR.2011.5995324