Angular-Driven Feedback Restoration Networks for Imperfect Sketch Recognition

Jia Wan, Kaihao Zhang, *Graduate Student Member, IEEE*, Hongdong Li, *Senior Member, IEEE*, and Antoni B. Chan, *Senior Member, IEEE*

***Abstract*— Automatic hand-drawn sketch recognition is an important task in computer vision. However, the vast majority of prior works focus on exploring the power of deep learning to achieve better accuracy on complete and clean sketch images, and thus fail to achieve satisfactory performance when applied to incomplete or destroyed sketch images. To address this problem, we first develop two datasets that contain different levels of scrawl and incomplete sketches. Then, we propose an angular-driven feedback restoration network (ADFRNet), which first detects the imperfect parts of a sketch and then refines them into high quality images, to boost the performance of sketch recognition. By introducing a novel "feedback restoration loop" to deliver information between the middle stages, the proposed model can improve the quality of generated sketch images while avoiding the extra memory cost associated with popular cascading generation schemes. In addition, we also employ a novel angular-based loss function to guide the refinement of sketch images and learn a powerful discriminator in the angular space. Extensive experiments conducted on the proposed imperfect sketch datasets demonstrate that the proposed model is able to efficiently improve the quality of sketch images and achieve superior performance over the current state-of-the-art methods.**

***Index Terms*— Imperfect sketch recognition, angular-based loss function, feedback restoration loop, attention module.**

I. Introduction

SKETCHES are a convenient way to express user intent. Naturally, the recognition of sketches becomes a popular topic due to its applications, such as human-computer interaction. In practice, most of the sketches are created by amateurs, rather than professionals. However, the current methods for sketch recognition are specialized for "perfect" sketches that are typically created by professionals or paraprofessionals. On the contrary, most amateur sketches from ordinary people usually exhibit several types of "imperfection". For example, the scale is not correct, some part is not completed, or some part is created with obliteration. Fig. 2 demonstrates the comparison between imperfect and perfect sketches. These imperfect sketches images lack important information, which cause current popular sketch recognition methods to not perform well.

To this end, we propose to address the problem of recognizing sketches with imperfection (we call them "imperfect sketches" for simplicity) via an additional restoration process. Directly recognizing imperfect sketches does not work well as these sketches have different types of imperfection artifacts, which are difficult to be modeled. We thus propose an end-to-end model to solve this problem by jointly performing restoration and recognition.

Specifically, given an imperfect sketch instance, we first feed it to a network whose aim is to restore its perfect counterpart that makes it easier to be recognized. The network is armed with a feedback mechanism, which conducts the comparison between the original imperfect sketch and the restored counterpart in a recursive manner. The difference will drive the next step of restoration, yielding a better restoration. Fig. 1 shows a comparison of different restoration schemes. Using a feedback loop that shares the network for each restoration iteration saves memory, especially when the iteration number increases. We also develop a new angular loss function that combines the advantages of the angular loss and the softmax loss, as imperfect sketch images are more difficult to be recognized than normal sketch images. Traditional recognition methods using softmax to learn features cause different margins for different classes. Specifically, the distances between inter-class features are large, while those of inter-class features are small. For imperfect sketch images, some important information may be missing. In order to extract more powerful discriminative features for these imperfect sketches, we apply the angular loss function, which defines the decision margin in the arc-cosine space. Additionally, we recognize imperfect sketch images by first restoring them. In order to reduce the distance between the restored sketch and real perfect sketch images, we apply adversarial loss to make the restored sketch images more realistic.

To verify the efficacy of our proposed method at recognizing imperfect sketches, we carry out extensive experimental studies. As there is not a suitable dataset which could be employed for our experiments, we thus develop two new datasets, containing sketches that are derived using SinGAN [1]. Marks or scratches are imposed on normal sketches to generate imperfect sketches. Ablation studies and comparisons with

Manuscript received July 2, 2020; revised February 23, 2021; accepted March 23, 2021. Date of publication April 15, 2021; date of current version May 21, 2021. This work was supported in part by the Research Grants Council of the Hong Kong Special Administrative Region, China under Project CityU 11215820, in part by the Australian Research Council (ARC) Centre of Excellence for Robotics Vision under Grant CE140100016, in part by the ARC-Discovery under Grant DP 190102261, in part by the ARC-LIEF under Grant 190100080, and in part by Baidu on autonomous driving. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Jing-Ming Guo. *(Corresponding author: Kaihao Zhang.)*

Jia Wan and Antoni B. Chan are with the Department of Computer Science, City University of Hong Kong, Hong Kong (e-mail: fjiawan1998@gmail.com; abchan@cityu.edu.hkg).

Kaihao Zhang and Hongdong Li are with the College of Engineering and Computer Science, The Australian National University, Canberra, ACT 2600, Australia (e-mail: fkaihao.zhang@anu.edu.au; hongdong.li@anu.edu.aug).

Digital Object Identifier 10.1109/TIP.2021.3071711

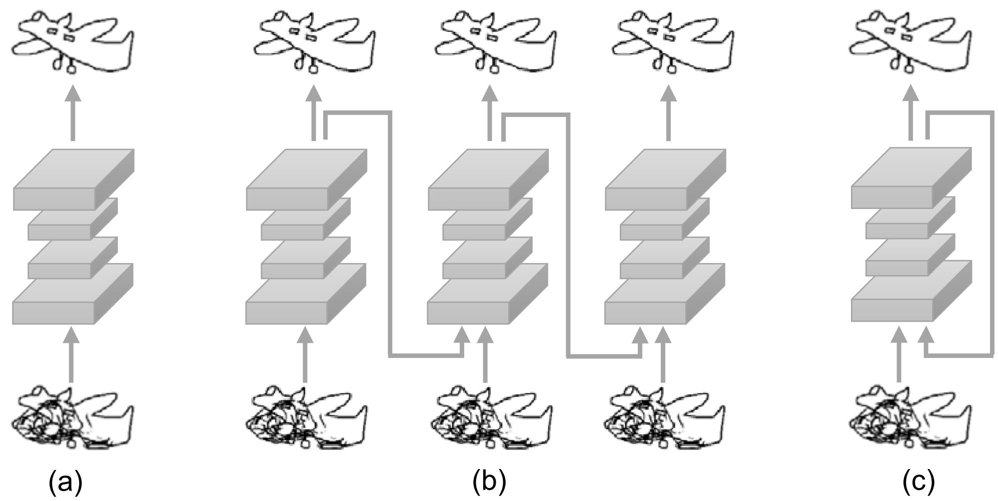

Fig. 1. **Different Restoration schemes.** (a). The direct restoration method. (b) The cascading architecture. (c) Our feedback restoration loop.

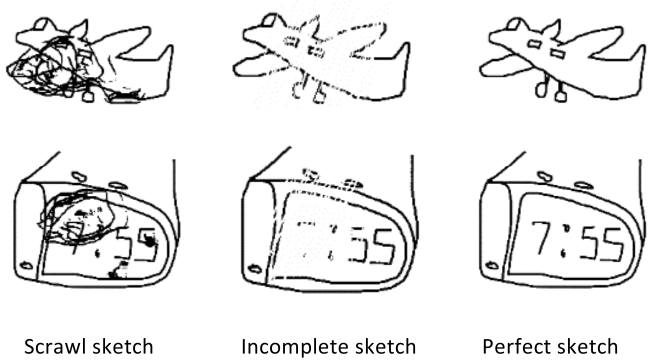

Fig. 2. **Examples of different imperfect sketches.** The two left sketches are scrawl sketches which are destroyed by unwanted slashes. The middle sketches are incomplete in which part of the sketch is erased. The right sketches are perfect sketches. Perfect sketches are much easier to be recognized.

other methods are conducted based on the new datasets, and the qualitative and quantitative results demonstrate the advantage of our method.

To summarize, our contributions are four-fold:

- From a practical perspective, we propose to address the problem of imperfect sketch recognition.
- We develop a deep neural network with a feedback mechanism to restore and recognize imperfect sketches at the same time. The restoration eases the difficulty of the recognition task, and the recognition task also benefits restoration via the use of semantic knowledge.
- A novel loss function based on angular penalty is derived to jointly recognize and restore imperfect sketches.
- We provide two Imperfect Sketch datasets, which are helpful to the community of sketch recognition.

II. Related Works

Our work is closely related to sketch recognition and image restoration, which are briefly discussed in the following.

A. Sketch Recognition

Different from recognizing natural images of great diversity, sketch recognition [2] is more difficult due to its abstract nature, and lack of colors and texture patterns. In this study we categorize sketch recognition methods based on whether deep learning is used or not. Typically, hand-engineered features are utilized as representations from which classifiers are trained. For instance, local hand-crafted features with BoW (bag-of-words) [3] is popular for sketch representation. Fisher vectors are utilized as the feature representation in [4]. Learning-based features are also used, e.g., multi-kernel learning is adopted in [5] to learn powerful features for free-hand sketch recognition from a pool of local features. Similarly, Yanık and Sezgin [6] show the advantages of active learning when applied to sketch recognition. Inspired by the seminal shape context descriptor, a new feature called Symmetric-aware Flip Invariant Sketch Histogram (SYMFISH) is proposed in [7] for representing sketches. In terms of classifiers, support vector machines (SVM) are a popular practical choice, such as the one used in [8]. Other options include nearest neighbour search, e.g., [9].

With the rise of deep learning, various approaches based on deep learning have been proposed recently [10]–[12]. The siamese network is popular for matching tasks. Thus in [13], a variant of Siamese networks is developed for matching natural images and sketch images. A triplet ranking model is proposed with corresponding data augmentation in [14]. The idea of hashing is extended from natural images to sketch images in [15], and an architecture is derived for encoding sketch images. Natural images are employed in [16] and a coarse-to-fine scheme is used. Similarly, Zhang *et al.* [17] develop a cousin network to guide sketch recognition under the help of natural images.

All the above works focus on traditional perfect sketch recognition. Imperfect sketch images are also common and are more difficult to recognize than perfect sketch images, due to destruction by redundant slashes, or incomplete parts that lose some important structural information. SketchGAN [18] uses a cascaded Encode-Decoder network to recognize incomplete sketch images. However, they ignore the cases where sketch images are also destroyed by redundant slashes. Meanwhile, the cascaded network also causes heavy memory cost, increasing with the number of cascade modules. Finally, SketchGAN can only recover incomplete sketches but cannot classify sketches into semantic classes.

In contrast to SketchGAN, we first extend the incomplete sketches to imperfect sketches, which consists of incomplete and scrawl sketches. Then, we propose a novel feedback loop that avoids heavy memory consumption to recover the imperfect sketches. Finally, we jointly recover the imperfect sketches and classify them into semantic classes with a novel angular-driven loss function.

B. Image Restoration

Image restoration [19], [20] is a general concept of removing artifacts caused by various factors. In general, it includes several different tasks, like image deblurring [21], [22], deraining [23] and inpainting [24]. Our task of sketch restoration is mostly similar to the general task of image inpainting with the following difference: 1) we focus on sketch, and 2) we remove artifacts and try to complete the sketch. Thus we primarily review related works on image completion. Most of the existing works are about natural image completion or contour completion. For natural image completion, great success has been achieved in recent years. To handle each spatial location differently, Yu *et al.* [25] propose a gated convolution network. Additionally, a loss function called SN-PatchGAN based on GAN is proposed to stabilize the training process. Similarly, partial convolution is introduced in [26] to conduct the convolution operation in only the valid pixels (uncorrupted areas). The mask is also automatically updated for the next layer. An attention module is employed in [27] to better complete image samples with multiple holes in arbitrary locations. The core idea is to borrow the surrounding regions as reference to guide the completion, to avoid inconsistent predictions regarding the neighbourhood regions. With user input as free-form masks, sketches and color, the SC-FEGAN [28] additionally employs a style loss to force the generated large-area completion to be more realistic. To deal with the missing high-level context, Yeh *et al.* [29] propose to utilize prior and context loss terms to search for the nearest neighbour of the image to be completed. The nearest neighbour then helps to better infer the missing areas. In [30], patch synthesis is carried out in multiple scales, respecting both the image contents and textures, and with the aid of a classification network, high-frequency details are predicted.

In terms of contour completion, the seminal work [31] formulates contour grouping as a graph problem. A grouping criterion named untangling cycle is proposed to utilize the topological structure. Ming *et al.* [32], [33] propose to use a high-order CRF model to complete contours with better closure. The derived high-order problem is solved efficiently by transforming it into an integer linear program.

III. Method

In this section, we present the structure of our restore-to-recognize network, which simultaneously restores and recognizes the corrupted sketch, as well as the associated loss functions used to train the network.

TABLE I

FCN Layer Settings for Different Modules of the Proposed Method. ConvX-Y Indicates Convolution Layer With Filter Size X With Y Output Channels

	Attention Net
Layer 1	Conv3-32 + ReLU()
Layer 2	Conv3-16 + ReLU()
Layer 3	Conv3-8 + ReLU()
Layer 4	Conv3-1 + ReLU()
	Restoration Net
Layer 1	Conv3-32 + ReLU()
Layer 2	Conv3-8 + ReLU()
Layer 3	Conv3-8 + ReLU() + MaxPooling()
Layer 4	Conv3-8 + ReLU() + MaxPooling()
Layer 5	Deconv3-8 + ReLU()
Layer 6	Deconv3-8 + ReLU()
Layer 6	Conv3-1 + ReLU()
	Classifier
Layer 1	Conv3-4 + ReLU()
Layer 2	Conv3-16 + ReLU()
Layer 3	Conv3-32 + ReLU()
Layer 4	Conv3-128 + ReLU()
Layer 5	Linear-class# + Softmax()

A. Restore-to-Recognize Network

In principle, it is very difficult to simultaneously solve the problems of sketch restoration and sketch recognition simultaneously. We thus adopt a coarse-to-fine scheme, and solve them progressively. The principle of gradualism is implemented by the recurrent mechanism. Note that, the gradualism can also be implemented by cascading multiple sub-modules, which is adopted in SketchGAN [18], but this strategy results in heavy memory and resource cost. Considering this, a recurrent mechanism is a better choice.

The structure of our restore-to-recognize network consists of four components, i.e., attention module, restoration module, recognition module and a discriminator, as illustrated in Fig. 3. The input of our network is the imperfect sketch. Throughout the restoration stages, the imperfect sketch is converted to its predicted "perfect" version (ideally). This converted sketch is then forwarded to the recognition module for the final recognition task.

1) Attention Module: Let the imperfect sketch be represented as S_{in}. The attention module aims to tell which part/component of the concerned sketch is not perfect. The attention module receives the imperfect sketch as the input and predict the possible imperfect area. We can formulate the module as,

$$a = F_{att}(S_{in}), \tag{1}$$

where F_{att} represents the attention function and a is the attention map. The detailed Attention Net structure is summarized in Tab. I.

2) Feedback Restoration Module: The restoration module plays an important role in our network. Better restoration quality leads to less difficulties in recognizing the sketch. At the same time, a better recognizer yields better restoration quality. The attention map from the preceding recurrent attention module will guide the learning process of the restoration.

The learned attention map a will be concatenated with the imperfect sketch S_{in} and served as input to the

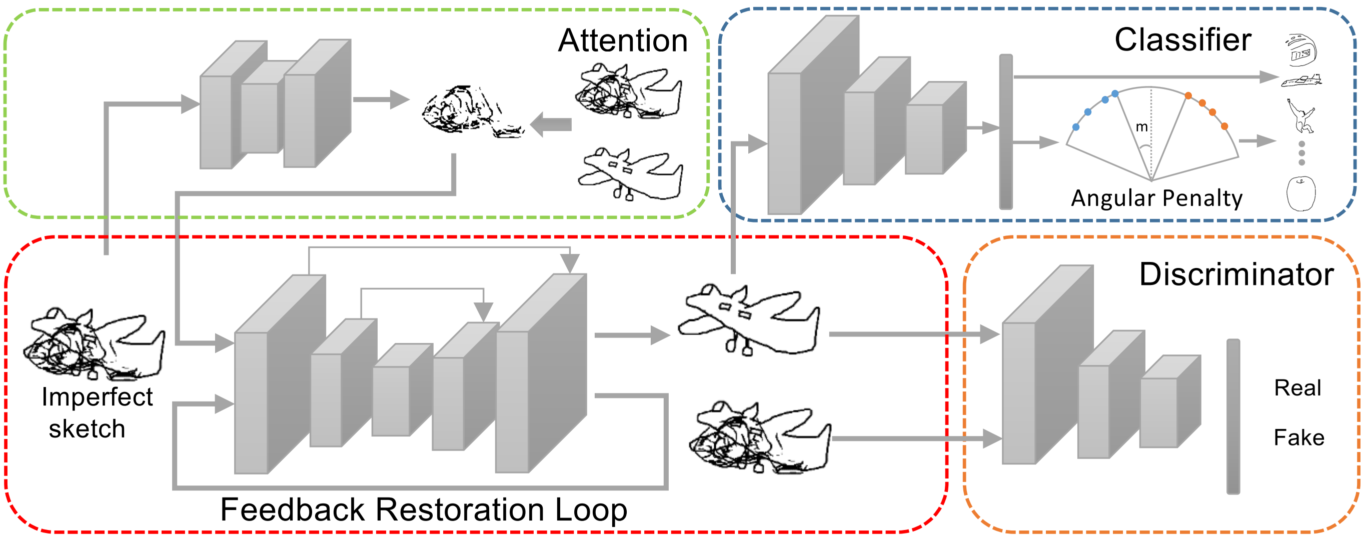

Fig. 3. The architecture of the proposed method. First, an attention module is used to discover the imperfect pixels. Then, a feedback restoration loop is used to restore perfect sketch based on the imperfect sketch and the attention. The discriminator is used for generating more realistic sketches. After the perfect sketch is generated, a angular-driven loss function is utilized to train the classifier on the restored sketches.

restoration module. The output of this module is the predicted "perfect" sketch. The first restoration step can be formulated as,

$$S_{rest}^{0}, f^{0} = F_{rest}(S_{in}, a, f^{init}), \tag{2}$$

where S_{rest}^{0} is the initial restored sketch and F_{rest} is the function of the restoration step, which is implemented as a neural network (Restoration Net) shown in Tab. I. f^{0} refers to the feedback information, which guides the subsequent restoration step. The initial feedback information f^{init} is zero. Then, the $i-$th feedback step can be calculate by:

$$S_{rest}^{i}, f^{i} = F_{rest}(S_{rest}^{i-1}, a, f^{i-1}), \tag{3}$$

where S_{rest}^{i} is the restored sketch from the i-th restoration step. The structure of the Restoration Net is shown in Fig. 4.

In addition to forwarding the predicted sketch S_{rest} to the following recognition module, S_{rest} is also sent to a discriminator D to distinguish it as real or fake imperfect sketch for adversarial training. The structure of the discriminator D is the same as the classifier in Tab. I.

3) Recognition Module: The recognition module accomplishes our ultimate goal of sketch recognition. The structure of the classifier is shown in Tab. I. However, due to the fact that sketches usually exhibit high-level abstraction and are absent from vivid colors and patterns, we force the recognition module to carry out two sub-tasks. One is the traditional multi-class classification, and the other one is to enforce both great inter-class difference and small intra-class difference. The latter one is learned by maximizing the additive angular margin, which drives the recognition module to learn more powerful features discriminating sketches with evident separability.

Accordingly, with the predicted perfect sketch S_{rest} from the restoration module as the input, there are two branches in the recognition module. The first one is a ordinary multi-class classifier, and the other one is also a multi-class classifier, but derived by the additive angular margin loss mentioned above.

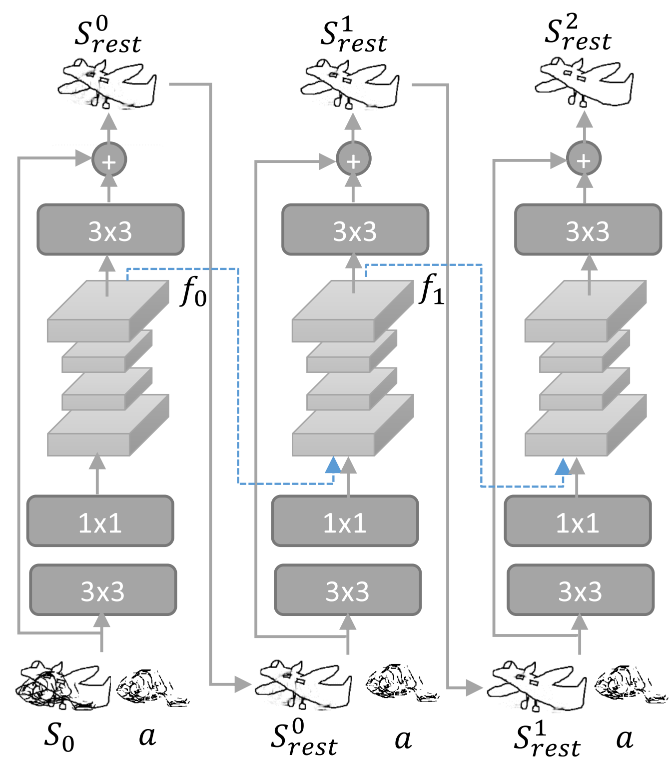

Fig. 4. **The architecture of the Feedback Restoration Loop.** The blue dotted line is the feedback flow.

B. Loss Function

In our restoration-to-recognize network, we employ the following loss function terms for training.

1) Attention Loss: The attention loss is used to supervise the learning of the attention module. The goal of the attention module is to discover the imperfection of the concerned sketch, therefore we expect the predicted attention is close to the ground truth locations of imperfection. Specifically, the attention loss is defined as

$$\mathcal{L}_{att} = ||a - a_{GT}||_2, \tag{4}$$

where we omit the superscript step index t to reduce clutter, and the ground truth of GT is defined based on the difference between clean/perfect sketch image S_{GT} and imperfect sketch S_{in},

$$a_{GT} = \text{thr}(S_{GT} - S_{in}), \tag{5}$$

where thr$(\cdot)$ means the element-wise threshold function.

2) Reconstruction Loss: With the attention and the corrupted sketch, we predicted a perfect version of the sketch. To enforce the restoration module to generate the expected sketch, we use the L2 norm of the difference between the ground truth sketch and the generated sketch as a loss,

$$\mathcal{L}_{rec} = ||S_{rest} - S_{GT}||_2 \tag{6}$$

3) Adversarial Loss: The above reconstruction loss function only forces the generated sketch to be close to the ground truth in the pixel space. However, the generated sketch is also required to be realistic in terms of human perception, which involves high-level abstraction, this task will be accomplished by the discriminator D mentioned above. To ensure the realistic property, we take the restoration module as a generator G, and employ an adversarial loss in the training, which is defined as,

$$\begin{aligned}\mathcal{L}_{adv}(G, D) = \mathbb{E}_{S\sim p_{data}}[\log(D(S))] \\ +\mathbb{E}_{\hat{S}\sim p_G}[\log(1 - D(G(\hat{S})))],\end{aligned} \tag{7}$$

where S represents an instance from real-world sketch sample set, and $\hat{S}$ indicates corrupted sketch. G is trained to fool the discriminator D, and as training proceeds, an equilibrium will be achieved.

4) Additive Angular Margin Loss: The input into the classifier is a generated sketch, and the output is the predicted class. Define the features of the penultimate layer as x_i. Traditional methods use Softmax Loss:

$$\mathcal{L}_{cls} = -\frac{1}{N}\sum_{i=1}^{N}\log\frac{e^{w_{y_i}^T x_i + b_{y_i}}}{\sum_{j=1}^{k} e^{w_j^T x_i + b_j}}, \tag{8}$$

where N and k are the number of samples and number of classes, (x_i, y_i) are the features and ground-truth class, and w_i and b_i are weights and bias terms of the classifier.

In practice, we also want to minimize the distance between a class center and its within-class samples, and maximize the distance between a class center and samples not in the class. To achieve this, an angular penalty is used for regularization. First, the cosine distance θ_i^j between sample i and class center j is calculated as:

$$\cos\theta_i^j = \frac{c_j^T x_i}{\|c_j\|\|x_i\|}, \tag{9}$$

where c_j is the center of $j-$th class, which is learned during optimization. Then, we define an angular loss based on θ_i^j as:

$$\mathcal{L}_{ang} = -\frac{1}{N}\sum_{i=1}^{N}\log\frac{e^{\cos\theta_i^{y_i}}}{\sum_{j=1}^{k} e^{\cos\theta_i^j}}. \tag{10}$$

The angular loss can be interpreted as the Softmax Loss taking only the angle between the feature and the center into consideration. To decrease intra-class distance and increase inter-class distance, we add a margin m to the angle between the sample and its class center. The angular margin loss $\mathcal{L}_{am}$ is written as:

$$\mathcal{L}_{am} = -\frac{1}{N}\sum_{i=1}^{N}\log\frac{e^{\cos(\theta_i^{y_i}+m)}}{\sum_{j=1}^{k} e^{\cos\theta_i^j}}, \tag{11}$$

where m is the margin that can better separate different classes compared to traditional Softmax Loss as shown in Fig. 3.

5) Final Loss: The defined loss functions above play different roles for the final generation results. Specifically, the attention loss aims to focus on the imperfect part of the sketch, the reconstruction loss and the adversarial loss improve the quality of the generated sketch, and the Softmax and the additive angular margin loss ensure learning powerful features for recognition. We utilized them in a weighted fashion to obtain the final loss function,

$$\mathcal{L}_{final} = \mathcal{L}_{rec} + \alpha\mathcal{L}_{att} + \beta\mathcal{L}_{adv} + \gamma\,\mathcal{L}_{cls} + \lambda\mathcal{L}_{am}, \tag{12}$$

where α, β, γ, and λ are hyper-parameters for weighting different loss functions.

IV. Experiments and Results

To evaluate the effectiveness of the proposed method, we conduct extensive experiments on scrawl and incomplete sketch recognition problems. We denote our method as ADFRNet, angular-driven feedback restoration networks.

A. Dataset and Evaluation Metrics

1) Dataset: Traditional methods mainly focus on perfect (noise-free) sketch recognition, while some sketches in real-world are imperfect (e.g., contain scrawl or incomplete lines). Therefore, we construct imperfect sketch datasets to enable research on imperfect sketches recognition. We use SinGAN [1] and data augmentation to improve the diversity of the generated imperfect sketches. The dataset can also be used to evaluate the robustness of algorithms since the imperfect sketches can be seen as noisy versions of the original sketches.

We generate corrupted and incomplete datasets based on a large-scale Sketch Database [34]. The database is consist of 75,471 sketch images with 125 classes. To generate the scrawl sketches, we first create 16 slashes by hand and then train 16 SinGAN [1] models with them. Each trained SinGAN model can generate similar slashes automatically. In our experiments, we generate 50 images for each trained SinGAN model, which results in 800 different slashes in total. Finally, we generate the scrawl sketches by randomly selecting one slash and adding it to the original sketch. To increase the diversity, we also use data augmentation on slashes using random rotation and resizing. For incomplete images, we remove the corresponding pixels in the original images. Typical corrupted and incomplete images can be seen in Fig. 2. To better evaluate the robustness of the proposed method, we further generate imperfect sketches with different scrawl or incomplete levels by repeating the generation process for multiple times. Different scrawl and incomplete level sketches can be seen in Figs. 6 and 7.

2) Evaluation Metrics: To evaluate the quality of the recovered sketches, we use peak signal-to-noise ratio (PSNR) and structural similarity (SSIM) to test the similarity at the pixel level. Higher PSNR and SSIM mean that the recovered sketches are closer to the original perfect versions. In some cases, it is not necessary that restored images with high reconstruction values are more easily recognized. Therefore, we use classification accuracy as the metric to further evaluate the recognition performance. If the accuracies of recovered sketches are higher than those of the imperfect versions, it means that the proposed ADFRNet both improves the qualitative performance of sketches and also benefits recognition of these imperfect sketches.

B. Ablation Study

To evaluate the effectiveness of different components in the proposed method, we compare the performance of six baseline networks via removing one of the components.

- **No attention.** This network is a version of ADFRNet without the attention module. The attention module is able to guide the ADFRNet to focus on the imperfect parts of sketches, and thus help recover the perfect version.
- **No classifier.** This network is a version of ADFRNet without the classifier. The label of sketches may be changed during the recovering process. In order to generate a high-quality sketches and avoid changing their labels, the classifier is applied in the ADFRNet to provide the necessary supervision.
- **No feedback.** This network is a version of ADFRNet without the feedback restoration loop. A one-stage restoration network is difficult to recover a high-quality sketch, thus some methods develop multi-stage networks, which improve the performance of restoration but increase the parameters of networks. We develop a feedback restoration loop to modify the recovered sketches without stacking more sub-networks.
- **No angular-driven.** This network is a version of ADFRNet without the angular loss function. This loss function is able to improve the ability of classifier in the proposed ADFRNet.
- **Direct recognition.** This network is a simple classifier without the process of restoration. In this way, the imperfect sketches are directly fed into the classifier for the prediction.
- **The whole ADFRNet.** This network is our whole ADFRNet. It takes an imperfect sketch as input to firstly detect the imperfect parts via an attention model. Then the feedback restoration loop is able to recover the sketch with multiple stages. The recovered images are finally fed into a classifier for recognition and a discriminator to make them more realistic.

TABLE II

Ablation Study on Scrawl Sketch Recognition Task. Direct Recognition is Classification Without Restoration

	Accuracy	PSNR	SSIM
no attention	73.43%	27.77	0.9494
no classifier	72.98%	37.35	0.9968
no feedback	75.79%	31.83	0.9747
no angular-driven	73.73%	37.88	0.9969
direct recognition	69.53%	-	-
ours	**77.17%**	**39.31**	**0.9978**

TABLE III

Comparison of Sketch Recognition With Different Scrawl Level

	Accuracy	PSNR	SSIM
level 1	77.14%	39.31	0.9978
level 2	76.04%	33.29	0.9837
level 3	75.00%	31.24	0.9877

The experimental results are shown in Tab. II. First, direct recognition based on scrawl images does not work because of the low-quality of the input images. Without attention or feedback, the PNSR and SSIM are decreased, which shows that the two components are essential to restore high-quality sketch images. The classification accuracy drops dramatically without angular penalty, which confirms its effectiveness. Our full model achieves the best classification performance and the quality of the restored images is also the best.

To better understand the quality of restored images of different models, we visualize a few examples in Fig. 5. Without attention or feedback restoration loop, the visual quality is decreased, which further confirms the effectiveness of those components. For the model without classifier or angular penalty, the quality of the restored images are similar to the full models which means that those two components have less impact on the restored quality. Nonetheless, there still exist small "smudging" artifacts in the "w/o angular" results compared to the full model, which is reflected in its lower PNSR/SSIM in Table II.

To evaluate the robustness of the proposed scrawl sketch recognition method, we first generate 3 different scrawl levels by adding 1, 2, or 3 random slashes to the original sketches. Then, we use our full model to perform the experiment and the results are shown in Tab. III and Fig. 6. With the increase of scrawl level, both restored quality and classification accuracy are decreased. However, as shown in Fig. 6, the proposed model can successfully restore the sketch even when the sketch is severely obfuscated, which confirms the proposed method is robust to different scrawl level.

C. Comparison With State-of-the-Art Methods

We next compare the proposed method with different state-of-the-art models for reconstruction and recognition of scrawl sketches, incomplete sketches, mixed scrawl/incomplete sketches, and real incomplete sketches.

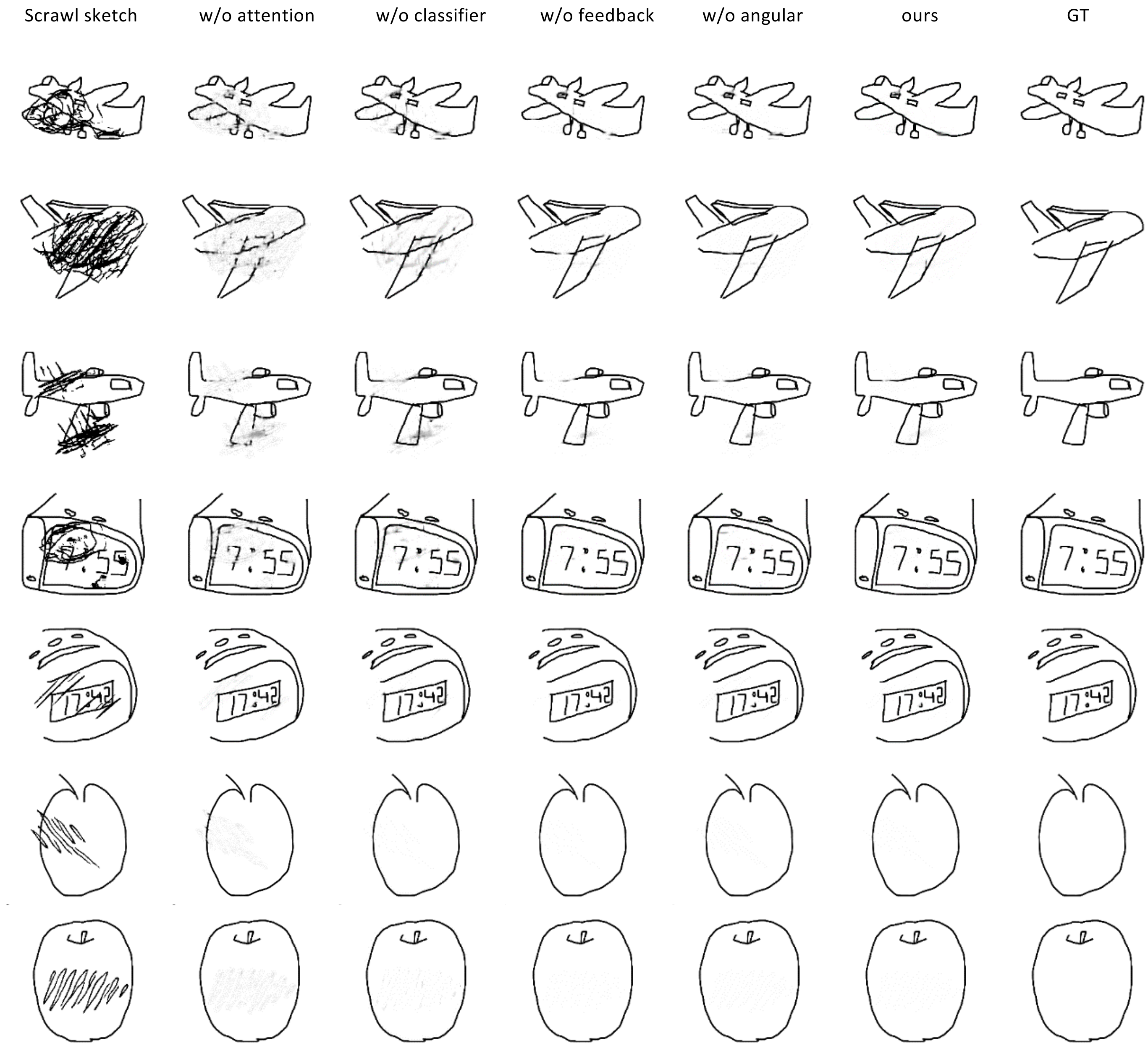

Fig. 5. Comparison of restored sketches from different models. From left to right, columns are scrawl sketches and restored sketches generated by different models. The last column is the ground truth sketches.

1) Scrawl Sketch Recognition: We first compare on scrawl sketch recognition. The first two comparison methods are two popular sketch classifiers Sketch-a-net [10] and Sketch-object-recognition [35]. For each classifier, we also use perfect images as training examples since the proposed method use them as the ground-truth during training. In addition, we compare two completion and recognition models: Generative image inpainting (Generative) [27] and SketchGAN [18]. Those comparison methods are summarized as follows:

- Sketch-a-net classifier [10]: Only trained on imperfect images.
- Sketch-a-net classifier [10]: Trained on both imperfect images and perfect images
- Sketch-object-recognition [35]: Only trained on imperfect images
- Sketch-object-recognition [35]: Trained on both imperfect images and perfect images
- completion+recognition: Generative image inpainting (Generative) [27], which is trained on both imperfect images and perfect images

TABLE IV
COMPARISON OF SKETCH RECOGNITION WITH DIFFERENT INCOMPLETE LEVEL

	Accuracy	PSNR	SSIM
level 1	76.51%	28.78	0.9898
level 2	75.93%	26.62	0.9847
level 3	74.88%	25.27	0.9790

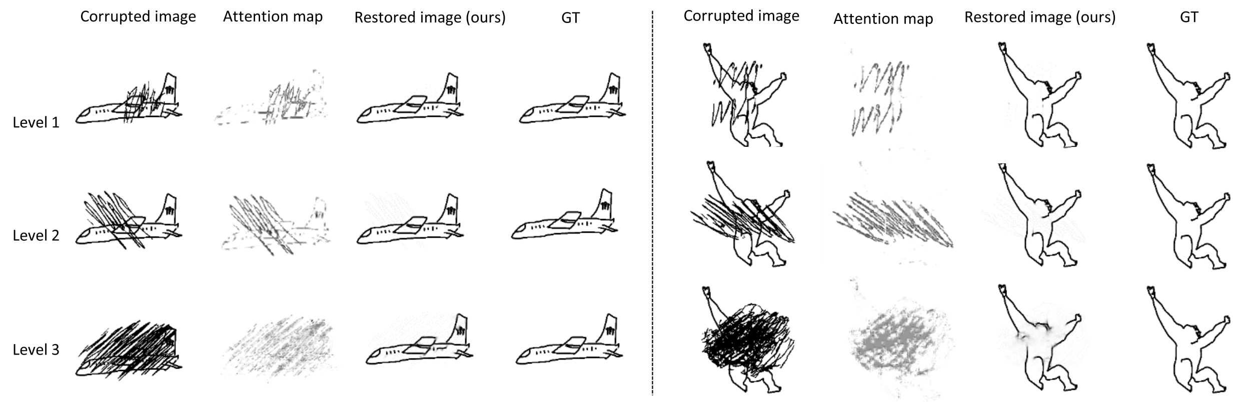

Fig. 6. Typical scrawl sketches with different scrawl levels, the corresponding restored sketches and the ground truth perfect sketches.

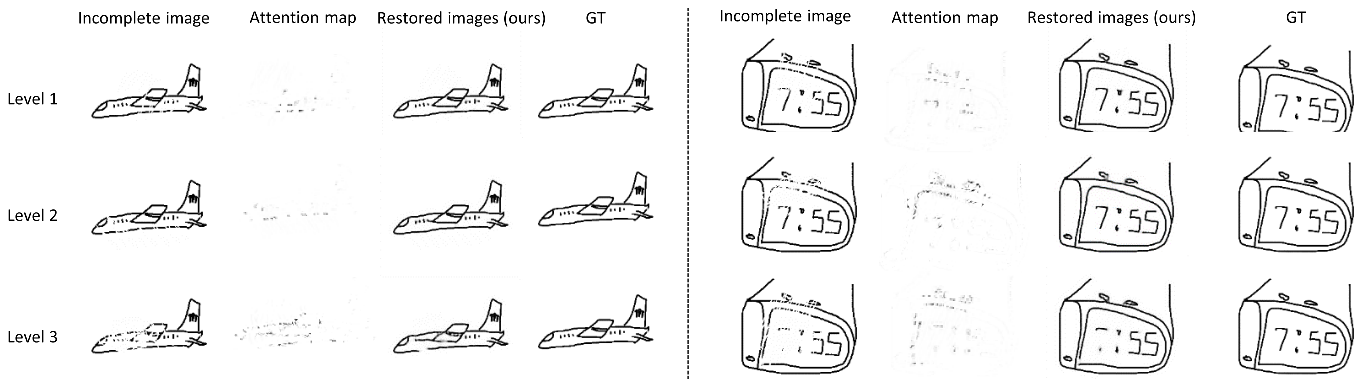

Fig. 7. Visualization of incomplete sketch restoration with different incomplete levels.

TABLE V
COMPARISON WITH STATE-OF-THE-ART MODELS FOR SCRAWL SKETCH RECOGNITION TASK

		Accuracy	PSNR	SSIM
Sketch-a-net [10]	Imperfect images	69.53%	-	-
	Imperfect + perfect images	74.78%	-	-
Sketch-object-recognition [35]	Imperfect images	71.23%	-	-
	Imperfect + perfect images	75.81%	-	-
Generative [27]	Imperfect + perfect images	75.71%	27.75	0.9844
SketchGAN [18]	Imperfect + perfect images	75.83%	29.44	0.9617
Ours	Imperfect + perfect images	**77.14%**	**39.31**	**0.9978**

- completion+recognition: SketchGAN [18], which is trained on both imperfect images and perfect images

Tab. V shows the quantitative performance of different models. Because the methods of Sketch-a-net [10] and Sketch-object-recognition [35] directly recognize the imperfect images without the restoration process, PSNR/SSIM cannot be computed and only the classification accuracy is provided. Training models on both imperfect and perfect images can improve the robustness of these models and thus achieve better performance on imperfect sketch recognition. Generative image inpainting [27] and SketchGAN [18] firstly recover the imperfect sketches and then make the prediction. Although the training samples are the same as [10] and [35], the group of completion+recognition achieve a better accuracy on sketch recognition. The proposed ADFRNet outperforms the current state-of-the-art methods in terms of accuracy, PSNR and SSIM, which demonstrates its effectiveness. Compared to direct classification approaches, the restoration process is useful for imperfect sketch recognition since noise contained in sketches will hurt the performance. Compare to "Generative" and "SketchGAN", the proposed method achieves better PSNR and SSIM as shown in Tabs. V and VI. With better sketches restored by a feedback loop, the overall classification performance of the proposed method is improved.

In order to qualitatively compare the ADFRNet with current methods, we show the restored images of different SOTA models in Fig. 8. The restored images generated by our proposed method are better than the other two models, which confirms the effectiveness of the proposed attention module and restored feedback loop.

2) Incomplete Sketch Recognition: Next, we evaluate the proposed model on incomplete sketch recognition task. We conduct experiments on an incomplete sketch dataset. Similar to scrawl sketch recognition, we use two classifiers and two completion and recognition models for comparison. As shown in Tab. VI, the proposed model achieves the best

TABLE VI

COMPARISON WITH STATE-OF-THE-ART MODELS FOR INCOMPLETE SKETCH RECOGNITION TASK

		Accuracy	PSNR	SSIM
Sketch-a-net [10]	Imperfect images	73.52%	-	-
	Imperfect + perfect images	74.85%	-	-
Sketch-object-recognition [35]	Imperfect images	74.83%	-	-
	Imperfect + perfect images	75.77%	-	-
Generative [27]	Imperfect images	70.29%	23.15	0.9611
SketchGAN [18]	Imperfect images	67.26%	27.44	0.9865
Ours	Imperfect images	76.51%	28.78	0.9898

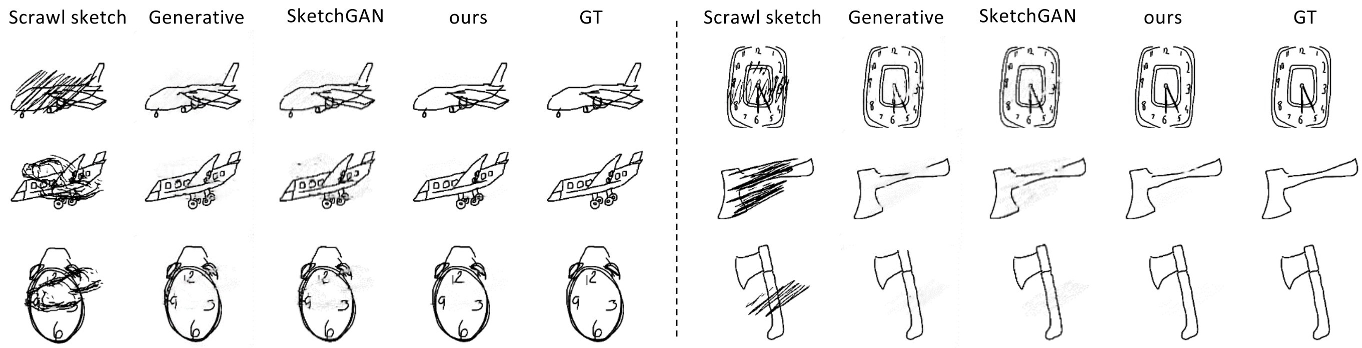

Fig. 8. Comparison with state-of-the-art models on scrawl sketch recognition tasks.

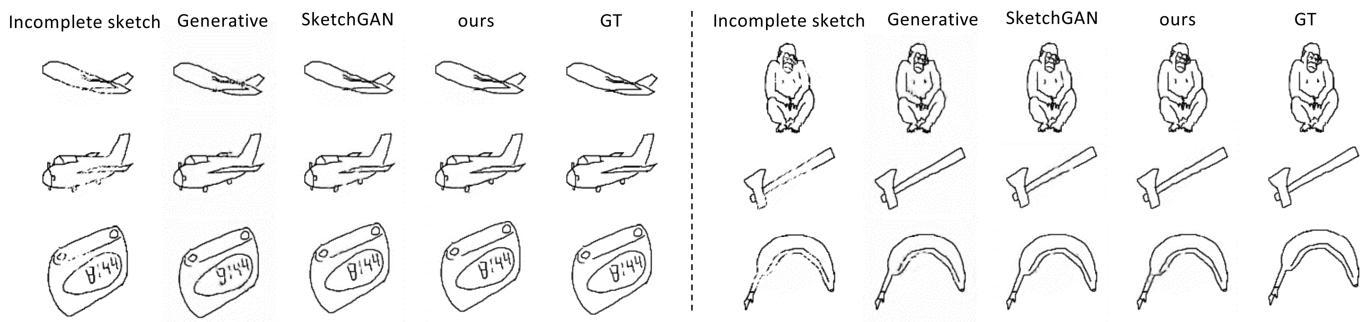

Fig. 9. Comparison with state-of-the-art models on scrawl sketch recognition tasks.

performance, which shows our method is also effective for incomplete sketch recognition task. Note that, the PSNR and SSIM of the proposed method is better than other completion methods and the generated images are more clear than comparison methods as shown in Fig. VI. This confirms the proposed method is effective at restoring sketch from incomplete images.

Similar to scrawl sketch recognition, an experiment with different incomplete levels is conducted to confirm the robustness of the proposed method. As shown in Tab. IV, the performance decreases as the level of incompleteness increases. Nonetheless, the proposed method is robust and completes the missing pixels as shown in Fig. 7.

3) Mixed Sketch Recognition: To evaluate the performance of general imperfect sketch recognition, we compare the performance of different approaches on a mixed dataset containing both scrawl and incomplete sketches. As shown in Tab. VII, our method achieves the best performance compared to other approaches, which confirms that the proposed method can be extended to a more general case. In addition, the restoration process generally improves the accuracy compared to direct classification methods.

TABLE VII

COMPARISON OF DIFFERENT APPROACHES WITH MIXED IMPERFECT SKETCHES

	Accuracy	PSNR	SSIM
Sketch-a-net [10]	68.84%	-	-
Sketch-object-recognition [35]	73.71%	-	-
Generative [27]	75.52%	26.23	0.9621
SketchGAN [18]	76.66%	26.99	0.9664
Ours	76.96%	27.63	0.9744

4) Real Imperfect Sketches: To evaluate the generalization ability of the proposed method, we conduct an experiment on a real-world dataset that is constructed by slashes drawn by hand instead of SinGAN. The models are trained with the synthetic dataset and evaluated on the real-world dataset. The performance is shown in Tab. VIII. Different from the previous results, the classification method "Sketch-object-recognition" achieves better performance than "Generative"

TABLE VIII
COMPARISON OF DIFFERENT APPROACHES ON REAL IMPERFECT SKETCHES

	Accuracy	PSNR	SSIM
Sketch-a-net	70.21%	-	-
Sketch-object-recognition	77.00%	-	-
Generative	70.76%	23.07	0.9478
SketchGAN	75.61%	29.14	0.9750
Ours	77.53%	36.28	0.9915

and "SketchGAN", which demonstrates that the restoration process can overfit. The proposed method still achieves better performance than "Sketch-object-recognition", which shows that the feedback loop generalize well to the real-world data compared to the other restoration approaches.

V. CONCLUSION

In this paper, we propose the problem of imperfect sketch recognition, which aims to solve two tasks: scrawl sketch recognition and incomplete sketch recognition. We generate two datasets with different scrawl and incompleteness levels. Finally, we propose a unified framework for those two tasks. The imperfect sketch is first restored by the proposed attention-based feedback restoration loop, and then sent to a classifier, which is trained using an angular-driven classification loss function. Extensive experiments confirm the effectiveness of the components of the proposed ADFRNet, and demonstrate that it achieves the state-of-the-art performance on imperfect sketch recognition tasks.

ACKNOWLEDGMENT

The authors gratefully acknowledge the GPUs donated by NVIDIA Corporation.

REFERENCES

[1] T. R. Shaham, T. Dekel, and T. Michaeli, "SinGAN: Learning a generative model from a single natural image," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 4570–4580.

[2] M. Eitz, K. Hildebrand, T. Boubekeur, and M. Alexa, "Sketch-based image retrieval: Benchmark and bag-of-features descriptors," *IEEE Trans. Vis. Comput. Graphics*, vol. 17, no. 11, pp. 1624–1636, Nov. 2011.

[3] J. Sivic and A. Zisserman, "Video Google: A text retrieval approach to object matching in videos," in *Proc. IEEE Int. Conf. Comput. Vis.*, Oct. 2003, p. 1470.

[4] R. G. Schneider and T. Tuytelaars, "Sketch classification and classification-driven analysis using Fisher vectors," *ACM Trans. Graph.*, vol. 33, no. 6, pp. 1–9, Nov. 2014.

[5] Y. Li, T. M. Hospedales, Y.-Z. Song, and S. Gong, "Free-hand sketch recognition by multi-kernel feature learning," in *Proc. Comput. Vis. Image Understand. (CVIU)*, 2015, pp. 1–11.

[6] E. Yanık and T. M. Sezgin, "Active learning for sketch recognition," *Comput. Graph.*, vol. 52, pp. 93–105, Nov. 2015.

[7] X. Cao, H. Zhang, S. Liu, X. Guo, and L. Lin, "SYM-FISH: A symmetry-aware flip invariant sketch histogram shape descriptor," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2013, pp. 313–320.

[8] Y. Li, Y.-Z. Song, and S. Gong, "Sketch recognition by ensemble matching of structured features," in *Proc. Brit. Mach. Vis. Conf. (BMVC)*, 2013, pp. 35.1–35.11.

[9] M. Eitz, J. Hays, and M. Alexa, "How do humans sketch objects?" *ACM Trans. Graph.*, vol. 31, no. 4, pp. 1–10, Aug. 2012.

[10] Q. Yu, Y. Yang, Y.-Z. Song, T. Xiang, and T. Hospedales, "Sketch-a-net that beats humans," in *Proc. Brit. Mach. Vis. Conf. (BMVC)*, 2015, doi: 10.5244/C.29.7.

[11] Q. Yu, Y. Yang, F. Liu, Y.-Z. Song, T. Xiang, and T. M. Hospedales, "Sketch-a-net: A deep neural network that beats humans," *Int. J. Comput. Vis.*, vol. 122, no. 3, pp. 411–425, 2017.

[12] S. Ouyang, T. M. Hospedales, Y.-Z. Song, and X. Li, "ForgetMeNot: Memory-aware forensic facial sketch matching," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 5571–5579.

[13] F. Wang, L. Kang, and Y. Li, "Sketch-based 3D shape retrieval using convolutional neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 1875–1883.

[14] Q. Yu, F. Liu, Y.-Z. Song, T. Xiang, T. M. Hospedales, and C. C. Loy, "Sketch me that shoe," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 799–807.

[15] L. Liu, F. Shen, Y. Shen, X. Liu, and L. Shao, "Deep sketch hashing: Fast free-hand sketch-based image retrieval," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 2862–2871.

[16] H. Zhang, S. Liu, C. Zhang, W. Ren, R. Wang, and X. Cao, "SketchNet: Sketch classification with Web images," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 1105–1113.

[17] K. Zhang, W. Luo, L. Ma, and H. Li, "Cousin network guided sketch recognition via latent attribute warehous," in *Proc. Assoc. Advancement Artif. Intell. (AAAI)*, 2019, pp. 9203–9210.

[18] F. Liu, X. Deng, Y.-K. Lai, Y.-J. Liu, C. Ma, and H. Wang, "SketchGAN: Joint sketch completion and recognition with generative adversarial network," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 5830–5839.

[19] Y. Gou, B. Li, Z. Liu, S. Yang, and X. Peng, "CLEARER: Multi-scale neural architecture search for image restoration," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 33, 2020, pp. 17129–17140.

[20] Y. Lin, Y. Gou, Z. Liu, B. Li, J. Lv, and X. Peng, "COMPLETER: Incomplete multi-view clustering via contrastive prediction," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021.

[21] K. Zhang *et al.*, "Deblurring by realistic blurring," 2020, *arXiv:2004.01860*. [Online]. Available: http://arxiv.org/abs/2004.01860

[22] K. Zhang, W. Luo, Y. Zhong, L. Ma, W. Liu, and H. Li, "Adversarial spatio-temporal learning for video deblurring," *IEEE Trans. Image Process.*, vol. 28, no. 1, pp. 291–301, Jan. 2019.

[23] W. Yang, R. T. Tan, J. Feng, J. Liu, Z. Guo, and S. Yan, "Joint rain detection and removal via iterative region dependent multi-task learning," *CoRR, abs/1609.07769*, vol. 2, no. 3, pp. 1–12, 2016.

[24] J. Xie, L. Xu, and E. Chen, "Image denoising and inpainting with deep neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2012, pp. 341–349.

[25] J. Yu, Z. Lin, J. Yang, X. Shen, X. Lu, and T. S. Huang, "Free-form image inpainting with gated convolution," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 4471–4480.

[26] G. Liu, F. A. Reda, K. J. Shih, T.-C. Wang, A. Tao, and B. Catanzaro, "Image inpainting for irregular holes using partial convolutions," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Sep. 2018, pp. 85–100.

[27] J. Yu, Z. Lin, J. Yang, X. Shen, X. Lu, and T. S. Huang, "Generative image inpainting with contextual attention," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2018, pp. 5505–5514.

[28] Y. Jo and J. Park, "SC-FEGAN: Face editing generative adversarial network with user's sketch and color," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 1745–1753.

[29] R. A. Yeh, C. Chen, T. Y. Lim, A. G. Schwing, M. Hasegawa-Johnson, and M. N. Do, "Semantic image inpainting with deep generative models," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 5485–5493.

[30] C. Yang, X. Lu, Z. Lin, E. Shechtman, O. Wang, and H. Li, "High-resolution image inpainting using multi-scale neural patch synthesis," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 6721–6729.

[31] Q. Zhu, G. Song, and J. Shi, "Untangling cycles for contour grouping," in *Proc. IEEE 11th Int. Conf. Comput. Vis.*, Oct. 2007, pp. 1–8.

[32] Y. Ming, H. Li, and X. He, "Contour completion without region segmentation," *IEEE Trans. Image Process.*, vol. 25, no. 8, pp. 3597–3611, Aug. 2016.

[33] Y. Ming, H. Li, and X. He, "Connected contours: A new contour completion model that respects the closure effect," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Providence, RI, USA. Washington, DC, USA: IEEE Computer Society, Jun. 2012, pp. 829–836, doi: 10.1109/CVPR.2012.6247755.

[34] P. Sangkloy, N. Burnell, C. Ham, and J. Hays, "The sketchy database: Learning to retrieve badly drawn bunnies," *ACM Trans. Graph.*, vol. 35, no. 4, pp. 1–12, Jul. 2016.

[35] R. K. Sarvadevabhatla, J. Kundu, and V. B. R, "Enabling my robot to play pictionary: Recurrent neural networks for sketch recognition," in *Proc. 24th ACM Int. Conf. Multimedia*, Oct. 2016, pp. 247–251.

Jia Wan received the B.Eng. degree in software engineering from Northwestern Polytechnical University, Xi'an, China, and the M.Phil. degree from the School of Computer Science, Center for Optical Imagery Analysis and Learning (OPTIMAL), Northwestern Polytechnical University, in 2015 and 2018, respectively. He is currently pursuing the Ph.D. degree in computer science with the City University of Hong Kong. His research interests include congestion analysis and crowd counting.

Kaihao Zhang (Graduate Student Member, IEEE) received the M.Eng. degree in computer application technology from the University of Electronic Science and Technology of China, Chengdu, China, in 2016. He is currently pursuing the Ph.D. degree with the College of Engineering and Computer Science, The Australian National University, Canberra, ACT, Australia. He worked with the Center for Research on Intelligent Perception and Computing, National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, Beijing, China, for two years, and the Tencent AI Laboratory, Shenzhen, China, for two years. His research interests include video analysis and facial recognition with deep learning.

Hongdong Li (Senior Member, IEEE) is currently a Professor with the Computer Vision Group, Australian National University (ANU). He is also a Chief Investigator with the Australia ARC Centre of Excellence for Robotic Vision (ACRV). Prior to 2010, he was with NICTA Canberra Laboratory working on the "Australia Bionic Eyes" Project. His research interests include 3D vision reconstruction, structure from motion, multi-view geometry, and applications of optimization methods in computer vision. He received a number of prestigious best paper awards in computer vision and pattern recognition, the CVPR Best Paper Award, in 2012, and the ICCV Marr Prize Honorable Mention, in 2017. He served as the Area Chair in recent year. He was the Program Chair for ACRA 2015—Australia Conference on Robotics and Automation and the Program Co-Chair for ACCV 2018—Asian Conference on Computer Vision. He is also an Associate Editor of IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE.

Antoni B. Chan (Senior Member, IEEE) received the B.S. and M.Eng. degrees in electrical engineering from Cornell University, Ithaca, NY, USA, in 2000 and 2001, respectively, and the Ph.D. degree in electrical and computer engineering from the University of California San Diego (UCSD), San Diego, in 2008. He is currently an Associate Professor with the Department of Computer Science, City University of Hong Kong. His research interests include computer vision, machine learning, pattern recognition, and music analysis.