Multimedia Tools and Applications (2024) 83:29047–29066
https://doi.org/10.1007/s11042-023-16704-z

Sketch-to-image synthesis via semantic masks

Samah S. Baraheem[1,2] · **Tam V. Nguyen**[2]

Received: 26 July 2022 / Revised: 23 May 2023 / Accepted: 27 August 2023 /
Published online: 9 September 2023

Abstract
Sketch-to-image is an important task to reduce the burden of creating a color image from scratch. Unlike previous sketch-to-image models, where the image is synthesized in an end-to-end manner, leading to an unnaturalistic image, we propose a method by decomposing the problem into subproblems to generate a more naturalistic and reasonable image. It first generates an intermediate output which is a semantic mask map from the input sketch through instance and semantic segmentation in two levels, background segmentation and foreground segmentation. Background segmentation is formed based on the context of the foreground objects. Then, the foreground segmentations are sequentially added to the created background segmentation. Finally, the generated mask map is fed into an image-to-image translation model to generate an image. Our proposed method works with 92 distinct classes. Compared to state-of-the-art sketch-to-image models, our proposed method outperforms the previous methods and generates better images.

Keywords Sketch-to-image generation · Sketch-to-image synthesis · Computer vision · Generative adversarial networks · Instance and semantic segmentation · Machine learning

1 Introduction

In the digital world and daily life, images are considered as one of the significant elements that can tell a long story. There is a famous saying, "A picture is worth a thousand words," meaning that complex and sometimes multiple ideas can be conveyed by a single still image. However, creating an image is not only a trivial task, but also time-consuming since it involves both low-level and high-level features such as colors, texture, brightness, and semantic information. Therefore, converting a sketch to its corresponding image has attracted attention from research community for many reasons. First, the sketch is composed of key structural information, where it contains only edges that shape the foregrounds which in turns can be useful to translate the sketch into a photo-realistic image. Moreover, the sketch does not include any pixel information which plays a significant role

✉ Samah S. Baraheem
ssbaraheem@uqu.edu.sa

[1] Department of Computer Science, Umm Al-Qura University, Al-lith, Saudi Arabia

[2] Department of Computer Science, University of Dayton, Dayton, OH, USA

in mapping the sketch into several naturalistic images in no time. Thus, the sketch-to-image synthesis helps artists, photographers, and animation makers reduce their repetitive workload. However, sketch-to-image synthesis is very challenging due to the lack of semantic information in the sketch and the complexity of the image. In addition, sketch is often highly sparse, abstract, and artist-dependent [71]. Further, the visual cues are lacked in the sketch [72]. Thus, to map a sketch, that contains key structural information, into an image, the research community has leveraged sketch-based image retrieval (*SBIR*) systems [1–4]. However, SBIR systems might face many problems. The system may not be able to retrieve fine-grained images. Furthermore, the system may not generate proper images in terms of the similarity to the input object sketch, in particular size, orientation, or occlusion aspects. Therefore, researchers have incorporated deep convolutional neural networks (*CNNs*) [5] to solve the sketch-to-image problem [6, 7]. While incorporating CNNs with the sketch-to-image task obtains better synthesized images than SBIR approach because the features are learned instead of being manually extracted, it gives unnaturalistic images with more than one sketched object. After generative adversarial network (*GAN*) [8] was proposed, the research community has shifted to GAN-based sketch to image [9–16] since it produces better results than before. However, it still struggles to produce photo-realistic images for complex scenes with multiple sketched objects.

To this end, this proposed research attempts to divide the sketch-to-image problem into subproblems to seamlessly create a naturalistic image from the sketch in four main stages. In the first stage, the objects are detected from the input sketch, and a mask is created for each detected object through instance segmentation models, for example, Mask R-CNN [17], Cascade RCNN [18], HTC [19], QueryInst [20], and DetectoRS [21]. In the second stage, for each created mask, a semantic segmentation process is performed to label each mask along with creating a semantic segmentation background. In the third main stage, the generated semantic segmentation map is fed into an image-to-image translation model, i.e., SPADE model [22] to generate the corresponding image. Following this step, a post-processing stage is implemented to enhance the synthetic image further through background improvement and face refinement. A sample of our sketch-to-image results is shown in Fig. 1. Our contribution is as follows.

- We first create a sketch-like image dataset based on MS-COCO [37] through four different methods, in particular, dodging and burning [38], Holistically-nested edge detection [39], Canny edge detector [40], and Sobel operators [41].
- Following this, we re-train several instance segmentation models, namely, Mask R-CNN [17], Cascade RCNN [18], HTC [19], QueryInst [20], and DetectoRS [21] on our newly generated sketch-like image dataset.

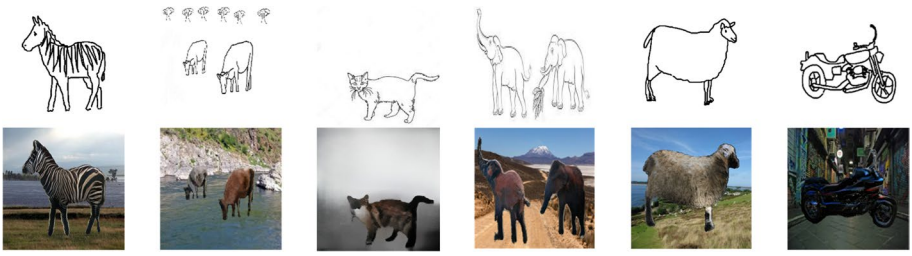


Fig. 1 A sample of our sketch-to-image results. The first row corresponds to the input sketch, while the second row shows the synthesized images generated by our framework

– Moreover, the semantic mask segmentation is implemented in two levels: background segmentation and foreground segmentation.
– In addition, two post-processing steps are incorporated to further enhance the generated images. During the background improvement step, a scene dataset of 10 k images categorized into 365 different scene classes is collected. Then, based on the classified scene, our framework selects a scene image and blends the foregrounds into the chosen background. Further, the face refinement step is implemented by extracting human faces, reconstructing them through image completion model, i.e., ICT model [45], and aligning them in the corresponding generated image.
– Last but not least, an evaluation dataset is collected. It contains 378 different sketch styles.

2 Related work

Generative AI is the means that uses AI algorithms to create content. Specifically, it generates an output from the data they are trained on. For instance, it synthesizes text, images, videos, 3D renderings, code, just to name a few. Recently, image generation has attracted the research community. Many applications have been developed, such as natural image generation [1–4, 6, 7, 9–16, 49, 50, 52, 53, 60, 75, 76], art generation [80–83], face generation [73, 77–79], and adversarial example generation [84–88]. Image generation is the process that converts the inputs into images. Sketch to image task [1–4, 6, 7, 9–16, 49, 50, 52, 53, 60, 73] converts the rough and simple sketch to a detailed color image. To convert a sketch to its corresponding image, research has been conducted using SBIR systems [1–4]. Then, due to the limitation in SBIR, researchers have incorporated CNNs [5] to translate a sketch into an image [6, 7]. After proposing GAN [8], many researchers have leveraged GAN into a sketch-to-image problem [9–16]. The taxonomy of the related work discussed in this paper is illustrated in Fig. 2.

2.1 Sketch-based image retrieval (*SBIR*)

SBIR system utilizes a database or a search engine to query the input sketch and retrieve the closest image to the input sketch. To determine the most similar image to the input sketch, different descriptors [23–28] are used to extract the features. Features play a significant role in determining the matched image to the input sketch.

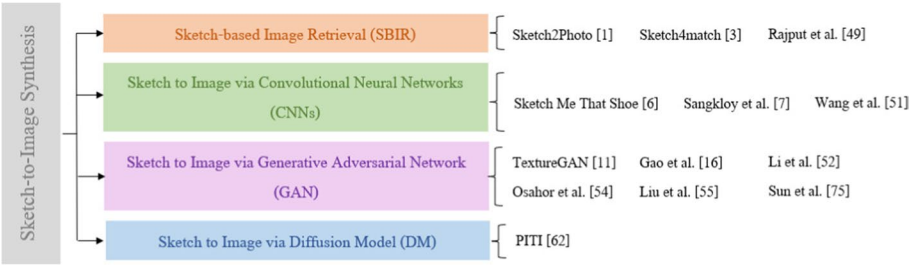


Fig. 2 The taxonomy of the related work discussed in this paper

Sketch2Photo [1] inputs the sketch along with the labels of objects in the sketch into an image search engine to retrieve an image for each object based on the provided label. Then, for each retrieved image, the image is segmented to determine the object that matches the sketched object. Next, the image composition step is required to blend all retrieved images through the blending technique. Following this, different results are evaluated based on the quality score to determine the best synthetic image. Due to using a search engine and an image composition, many problems could occur, i.e., improper object size, improper object orientation, or improper object occlusion. Furthermore, this method may produce artifacts, leading to unrealistic synthetic images especially with complex scenes.

Sketch4match [3] uses various descriptors [23, 24, 27] to extract the features of not only the sketch, but also the image after some preprocessing steps. These preprocessing steps on the image are required due to the fine-grained details in the color image and the simplicity of the sketch. While the extracted features on the color images are stored in a database, the features of the sketch are extracted when the sketch is entered into the system. A comparison between the feature vector of the sketch and the feature vectors of the images in the database is made to determine the closest image to the sketch. This step depends on Minkowski distance [29] and classification-based retrieval [30]. One major problem is that the system is implemented on a small database. Therefore, sometimes it is challenging to find the proper image that matches the input sketch.

To retrieve the best correct match of natural image to the corresponding input sketch through SBIR system, Rajput et al. [4] adopt a large dataset. A preprocessing step is required prior the feature extraction process from the sketch and natural image, where Otsu's approach is used to extract only the strong contours. Then, the features are extracted in two phases to extract the features from non-overlapping and 20% overlapping grids. Following this, a weighted similarity approach is leveraged, where the range [0,1] is used to apply weights to both overlapping and non-overlapping grids. Based on Euclidean distance [48], the most similar image to the input sketch is retrieved.

2.2 Sketch to image via convolutional neural networks (*CNNs*)

Due to the limitations in SBIR, researchers have shifted to leverage convolutional neural networks [5] to translate a sketch into an image. Additionally, since SBIR depends on extracting the features manually, it lacks the fine-grained retrieval. However, CNN depends on learning the features.

Sketch Me That Shoe [6] utilizes a new collected dataset of 1432 sketch-image pairs of only two labels (shoes and chairs). A deep convolutional neural network CNN, i.e., Siamese network [31] is used for the goal of triplet ranking. For this, three identical Sketch-a-Net [32] are leveraged. However, the network might overfit, meaning that the network is not able to generalize on new inputs that are not seen during the training process. This problem might happen since it is not enough to train a triplet ranking on only 1432 pairs.

Sangkloy et al. [7] use a larger new collected dataset of sketch-image pairs. The size of the dataset is 75,471 pairs, and the number of labels is 125 classes. The sketches are not simple, and they contain fine-grained details. Then, the dataset of sketch-image pairs is trained on convolutional neural networks CNNs such as AlexNet [33] and deep GoogLeNet [34] to retrieve objects of correct labels with fine-grained similarity to the sketch. Since the domain of sketches is different from the domain of images, the model is trained on sketches and images independently to embed their features separately. Although the dataset is large, the synthetic images lack a high level of realism, leading to unnaturalistic images.

Wang et al. [49] adopt two convolutional neural networks (CNNs) to improve sketch-based image retrieval (SBIR) results through re-ranking system. While the first CNN (Q-Net) is responsible for capturing the semantic information of the input sketch, the second CNN (N-Net) is responsible for capturing the category information of initial retrieved natural images that are obtained through initial SBIR system. Following this, a category similarity measurement approach is used to compute the similarity between the input sketch and initial retrieval results in terms of the semantic information extracted from softmax layer vectors of both networks. Then, the most similar natural images to the input sketch are determined via re-ranking process.

2.3 Sketch to image via generative adversarial network (*GAN*)

Due to inability to generate photorealistic images through CNNs, research has moved again to incorporate generative adversarial network GAN. TextureGAN [11] uses not only sketches as inputs, but also the color strokes and the texture; thus, the user has to select the color strokes and texture and place them on top of the sketch to determine the color and texture of each sketched object. This helps generate photorealistic images through GAN. Although this model works fine and generates naturalistic images, it is only trained on three labels which are handbags, shoes, and clothes; and thus, it is unable to generate images of other objects.

Gao et al. [16] generate images from input sketches in two levels instance-level and scene-level through two steps foreground generation and background generation. The first step is foreground generation which concentrates on generating the objects as provided in the input sketch. Thus, it uses a sketch segmentation approach [35] to detect and locate the objects in the input sketch. Then, it produces an image of only foreground objects. A background generation step is run by taking the generated foregrounds and the background sketch as inputs in the pix2pix model [36] to generate the final output image. One drawback of this method is that it is unable to synthesize realistic images from complex sketches with fine-grained details. Li et al. [50] introduce a two stage semi-supervised GAN-based image generation model. The first stage takes the class label and random noise as inputs and generates common information for every label learned via cGAN [51]. Next, during the second stage, a synthesized image is produced by leveraging the common information and the sketch via another cGAN model.

Osahor et al. [52] propose to generate multiple face images with various target attributes, such as gender, age, and hair color from a single input face sketch. The generator incorporates an identity preserving and quality guided networks. The quality guided encoder is utilized to improve the quality and decrease the dissimilarity between the synthesized image and its respective original image with regard to the embedding. To maintain the biometric identity of the synthetic image during the training process, the identity preserving network is leveraged. Then, a hybrid discriminator is used to infer a variety target attributes to generate different images with different attributes.

In another work, Liu et al. [53] eliminate the need of sketch-image pairs dataset by incorporating an unsupervised learning model to create different sketches for each image. Then, an auto-encoder [54, 55] is used along with a self-supervised approach [56, 57] and momentum mutual-information minimization loss [58] to separate the features into style and content features for both natural images and input sketches. The auto-encoder is composed of two encoders, style encoder and content encoder. While the style encoder generates a style feature map from the natural image, the content encoder produces a content feature map from the input sketch. Following this, the decoder generator takes the two produced feature

maps and generates an image. To ease extracting the style feature maps for fine-grained texture and distinct colors, the momentum mutual-information minimization is adopted.

In sketch-based image generation, most of the focus is on sketch-to-image synthesis of natural scenes. Some approaches are implemented to generate color face images from face sketches. One of the prior works that concentrates on learning the illumination over faces is proposed by Sun et al. [73] incorporating generative adversarial fusion model, namely, GAF. GAF integrates two U-Net generators and a single discriminator, where the illumination is learned and controlled by a parametric tanh (ptanh) activation function. This activation function is employed between the two generators through an illumination distribution layer. To preserve the identity and refine the facial details for fine-grained generated images, an attention mechanism is integrated into the second generator.

2.4 Sketch to image via diffusion model (DM)

Diffusion models [59], a specific form of generative model, have recently shown tremendous promise in generating high-quality images. As a result, several studies [60–67] in the field of image synthesis have taken advantage of diffusion models [59]. The forward diffusion phases of the diffusion model [59] start by slowly and progressively adding random noise to the input. The model then learns to reverse the diffusion process in order to reconstruct the input sample from the noise.

Pretraining-based Image-To-Image translation, often known as PITI [60], is a straightforward but efficient approach that employs a pretrained model to capture the whole distribution of the natural image. More specifically, GLIDE [68], a diffusion model, is adopted as a pretrained generative prior. A hierarchical generation mechanism [68–70] is employed to improve the synthetic image quality. This technique creates a coarse image initially, followed by super-resolution image. To enhance the diffusion model's texture generation, an adversarial training is also added throughout the denoising process. PITI [60] works successfully for translating geometry and masks into images; however, it might fail when mapping sketches into images, especially when various sketching styles are used as inputs. This could be attributed to the fact that the model was trained on sketch-like images obtained only by Holistically-nested Edge Detection (HED) [39].

3 Proposed method

In this section, we first introduce our dataset that is used. Then, our proposed methodology is discussed.

Dataset We aim to build an edge map dataset based on Microsoft COCO dataset [37] (over 118 k for training and 5 k for validation). MS-COCO [37] has 92 different classes, and it is considered a complex dataset because of its complexity level and multiplicity and difference of objects. Four methods are considered to synthesize edges from images. The first algorithm is dodging and burning [38] which considers two image blending techniques to maintain the structural details of the image content and obtain a pencil sketch output. The second algorithm is Holistically-nested edge detection [39] that obtains a coarse edge and minimal structural information. In addition, Canny edge detector [40] and Sobel operators [41] are used to obtain the edge map of COCO dataset.

Methodology The overall structure of the proposed method is illustrated in Fig. 3. The method consists of four main stages. In the fourth and last stage, two post-processing steps are implemented to enhance the results further via a background improvement and a face refinement steps. Details are discussed as follows.

Instance segmentation stage In this stage, the generated edge maps based on MS-COCO dataset [37] are used as inputs into an instance segmentation model. Different instance segmentation models are used, such as Mask R-CNN [17], Cascade RCNN [18], HTC [19], QueryInst [20], and DetectoRS [21]. For each instance segmentation model, the pretrained model on the COCO dataset of RGB images is utilized and fine-tuned on the edge maps dataset. The parameters are selected via preliminary studies. All instance segmentation models are fine-tuned and re-trained on our newly generated sketch-like images dataset for 30 epochs. For all models, the optimizer used during the training process is Stochastic Gradient Descent (SGD) with an intial learning rate of 0.02 and momentum of 0.9. One exception is with QueryInst [20], where AdamW is leveraged as an optimizer with learning rate of 0.0001. For all models, the weight decay is 0.0001. As for the backbone architecture, Mask R-CNN [17], Casecaded RCNN [18], and HTC [19] integrate ResNext-101 as the backbone architecture. In the meantime, for DetectoRS [21] and QueryIns [20] use ResNet-101 as the backbone architecture. The ultimate goal of this stage is to detect and locate 92 objects in the input sketch based on COCO objects, and then produce a mask for each object.

Semantic mask segmentation stage In this stage, the background is first segmented semantically based upon the detected objects in the first stage. For example, if the instance segmentation model recognizes three cars, five people, and one dog, the background is determined to be outdoor, in particular, a street scene. Thus, the background is segmented to sky and street. Specifically, to create a meaningful background based on the context, a simple algorithm is defined. First, an empty image with the same size as the input sketch is created. Then, to determine the background scene, we categorize each object in one or more background scenes based on the prior knowledge about where the object can be found in real world. Next, for each detected object from the previous stage, we increase the corresponding background scene(s) by one. The highest background scene is considered the actual background scene of the corresponding sketch input. In total, eight background scenes are identified. These background scenes are categorized into indoor and outdoor. For the indoor

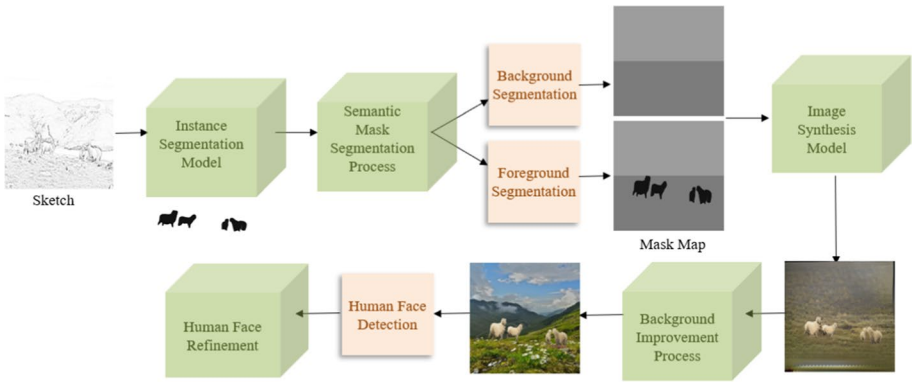


Fig. 3 The overview of our proposed framework

scenes, two scenes are specified based on the type of flooring, i.e., floor or carpet scenes. Meanwhile, for the outdoor scenes, six scenes are defined, namely, beach scene, park scene, snow scene, street scene, sidewalk scene, or sea scene. Therefore, after specifying the actual background scene, a semantic segmentation process is applied for the background, where for each and every pixel in the background, the pixel value is manipulated based on the determined background scene. Indeed, all scenes are divided into two segmented regions, except the beach scene. Thus, if the detected background is beach scene, the image is segmented into three regions which are sky, sea, and sand. Meantime, if the detected background is snow scene, the image is segmented into two regions which are fog and snow. Then, for each region, pixels are modified based on the region label, following COCO-Stuff [42] labels. Following this, foregrounds are segmented sequentially and inserted to the created segmented background. In particular, each mask is processed and segmented semantically with a particular value based on the class number assigned to each class of COCO classes. Then, every pixel that belongs to that mask has the same specific value.

Semantic mask to image synthesis stage Following the completion of the mask map, the third stage begins by taking the semantic mask map as input and feeding it to an image-to-image translation model. Specifically, the pre-trained SPADE model [22], that is trained on COCO-Stuff dataset [42], is leveraged in our proposed framework. As a result, this stage produces a synthetic image from an input semantic mask map.

Post-processing stage To enhance the synthesized images further, two post-processing steps are implemented as follows.

Background improvement step In the background improvement post-processing step, a scene classifier is used, namely, Places365-CNN [43] to recognize the scene from the synthesized images. Thus, for each generated image from the previous stage, the pre-trained scene classifier, i.e., Places365-CNN [43] is used to predict the scene category. There are 365 scene categories in Places365 [43]. After that, our system randomly selects a single scene image based on the classified scene and from our newly collected dataset.

Our scene dataset contains around 10,000 images categorized into 365 classes as Places365 [43], where each class has about 27 images. We collect our dataset using Google Image, where the scene should be empty without any distracting foregrounds to maintain our synthesized images. Following this step, the first two stages (instance segmentation stage and semantic segmentation stage) are performed on the input sketch image again with one difference. The background segmentation step is ignored and only the foregrounds are segmented, meaning that the background is empty, i.e., white background. This semantic mask map works as a binary image to help in segmenting and extracting the foregrounds from the background. Then, an alpha blending step is used to blend the chosen scene with the segmented extracted foregrounds. More specifically, this semantic mask map is resized to the same size as the synthesized image (256×256), followed by converting it to gray-scale color image. To segment the foregrounds from the background in the synthesized images, Otsu's thresholding [74] is integrated. Next, the thresholded image is converted back to RGB image, and a bitwise_and operator is used to visualize the segmented masked foregrounds. Subsequently, the black background resulted in previous step is changed into transparent background to ease the next step. Lastly, the extracted foregrounds are merged and blended into the chosen scene after resizing the scene into the same size which is 256×256. An illustration of the background improvement stage is shown in Fig. 4.

Face refinement step In the face refinement post-processing step, we adopt a face detector model [44] to detect any human faces in the synthesized images after the background improvement step. Then, each face region is extracted, and a binarized image of the same size as the extracted face region is created. In the binarized image, a square mask is created in the center. This helps to cover part of the extracted face region in order to reconstruct the face. Next, an image completion model, in particular, the ICT model [45] is utilized to reconstruct and complete the covered region of the face.

Following this step, a binarized image is created as the same size as the reconstructed face with a circle mask implemented. The reconstructed face and its corresponding mask are resized to have the same size as the extracted face. Following this, a bitwise_and operator is used to visualize only the reconstructed masked face region. Afterwards, the black background is changed into transparent background to help during the blending process. Therefore, this reconstructed face is blended onto the corresponding synthesized image and aligned in the same location. Face reconstruction stage is demonstrated in Fig. 5.

4 Experiments

4.1 Experimental settings

Dataset To evaluate our proposed method with state-of-the-art sketch-to-image methods, we compile a new dataset for testing purposes on different types of sketches. Since people have different sketching styles, we collect a dataset based on Sketchy dataset [7], Scketchy-COCO dataset [16], and via Google Image search engine. Fourteen classes (cat, dog, horse,

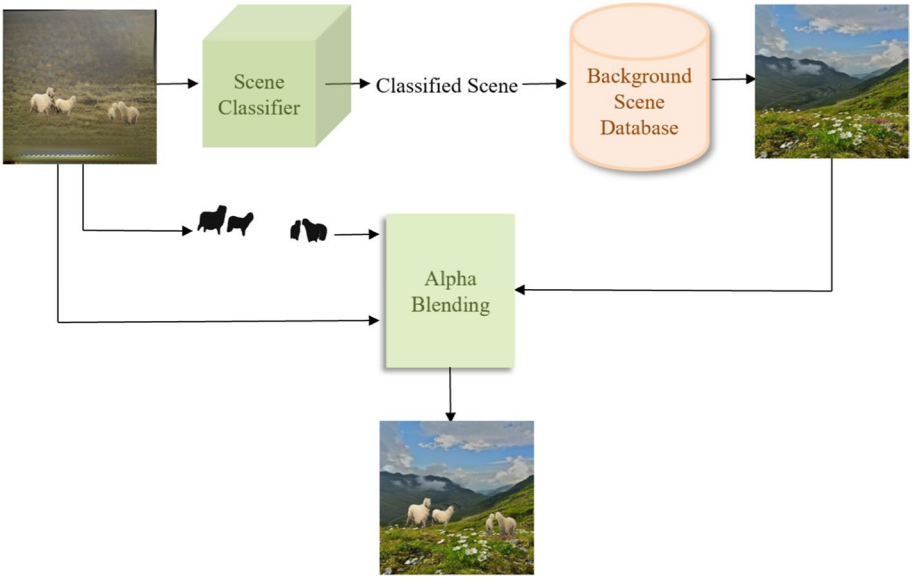


Fig. 4 An illustration of the background improvement stage. We adopt a scene classifier and the background scene database to obtain the relevant background scene

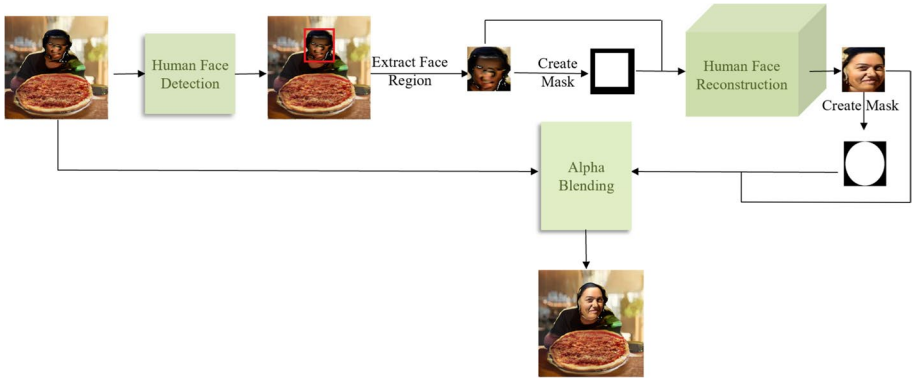


Fig. 5 A demonstration of face refinement stage. First, a human detector is used to detect the existing face(s) in the synthesize image. Then, each detected face is corrected to provide a clear face

sheep, cow, elephant, zebra, giraffe, car, bicycle, motorcycle, airplane, traffic light, and fire hydrant) are selected, where these classes belong to both aforementioned datasets along with MS-COCO dataset [37]. Our evaluation dataset consists of 378 sketches.

Baselines We compare our proposed method with the model proposed by Liu et al. [13], EdgeGAN [16], and PITI [60]. SketchyCOCO dataset [16] is selected to train the model proposed by Liu et al. [13]. Therefore, the original networks are trained on SketchyCOCO [16] through dodging and burning sketching techniques [38] to obtain pencil sketch images. Only the fourteen classes of SketchyCOCO are used in the training phase. Since fourteen classes are chosen, so fourteen individual models are trained separately for each class. However, this model generates an image into two steps, where the first step is to translate the sketch into grayscale image via shape translation network. Then, the second step is utilized to generate the color image from the generated grayscale image through a content enrichment network. Thus, we train the model with 400 and 200 epochs for step1 and step2, respectively, as suggested in the original work. Regarding EdgeGAN [16], the pre-trained model trained on SketchyCOCO dataset [16] is used to evaluate the model on our collected testing dataset. As for PITI [60], the pre-trained sketch-to-image model trained on COCO-Stuff dataset [42] is utilized.

Evaluation metrics To quantitatively evaluate the synthetic images, Fréchet Inception Distance (FID) [46] and Inception Score (IS) [47] are adopted. Both of these metrics attempt to assess the quality of the synthesized images. However, FID [46] is used to compare the distribution of synthetic images with the distribution of ground truth images, while IS [47] is utilized to evaluate the distribution of synthetic images based upon the realism, the quality, and the diversity of the generated images. IS metric [47] is used to judge synthetic images as human judgment. IS [47] is computed as follows.

$$IS = \exp\left(E_x \mathrm{D}_{KL}\left(\mathrm{p(y|x)} \,\middle\|\, \mathrm{p(y)} \right) \right) \tag{1}$$

where D_{KL} indicates Kullback-Leibler (KL) divergence, a measure that computes the similarity/difference between two probability distributions. In particular, the two distributions are the conditional probability distribution p(y|x) and the marginal probability distribution

p(y), where x refers to the generated sample, and y denotes the predicted label through the inception model.

As for FID [46], it considers both the generated and real image by computing the distance between the distribution of the generated image $p_g(x)$ and the distribution of the real image $p_{real}(x)$ in terms of the extracted features. The formula is illustrated in Eq. (2).

$$\begin{aligned} FID\left(p_{real}, p_g\right) &= d^2\left(\left(m_{real}, m_g\right), \left(m_g, c_g\right)\right) \\ &= \left\| m_{real}, -m_g \right\|^2 + T_r\left(c_{real} + c_g - 2\left(c_{real}\, c_g\right)^{1/2}\right), \end{aligned} \tag{2}$$

where d is the distance. m_{real}, c_{real}, m_g, and c_g are mean and covariance of real and generated images, respectively. T_r refers to the trace linear algebra operation, i.e., the summation of elements of the square matrix's main diagonal.

Since the real images (the ground truth) is required for calculating FID [46], we use this metric only during the ablation study of our work. Furthermore, a user study is conducted to assess how realistic the synthetic images are and how similar the generated images are to their corresponding sketches through computing the Average Human Rank (*HR*). Table 1 details the notations used in this paper.

Evaluation on sketches dataset We compare our proposed model with the baselines quantitatively and qualitatively based on sketches dataset that we collected to contain different types of sketches. In addition, an ablation study is conducted to analyze the effectiveness of some components of our proposed method.

4.2 Quantitative results

Realism and diversity factors are important in evaluating the quality of synthetic images; thus, we compare our method with the baselines based on IS [47]. Higher IS score indicates a better model with regard to quality and diversity of synthesized images. The comparison in terms of IS between our proposed method and previous methods is summarized in Fig. 6 and Table 2. For fairness purposes, the backgrounds are excluded from all generated images produced by our proposed framework to compare the results in only object-level as well as scene-level. As can be seen from Fig. 6

Table 1 Notations and their corresponding meaning

Notation	Meaning
D_{KL}	Kullback-Leibler divergence
$p(y\|x)$	Conditional probability distribution
p(y)	Marginal probability distribution
x	Generated image
y	Predicted label through the inception model
m_{real}	Mean of real image
m_g	Mean of generated image
c_{real}	Covariance of real image
c_g	Covariance of generated image
T_r	Trace linear algebra operation
HR	Average Human Rank

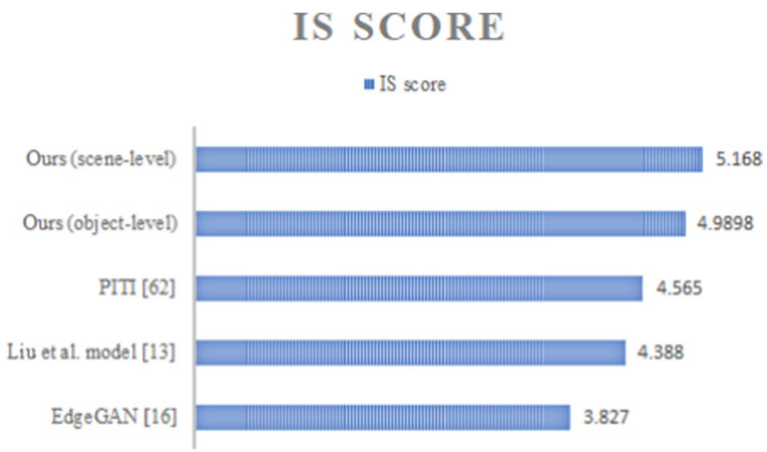


Fig. 6 An illustration of the mean of IS score [47] of our proposed framework and baselines [13, 16, 60]

and Table 2, our proposed method achieves the higher IS score in both object-level and scene-level, meaning that it is the best model in terms of quality and diversity. The reason is that the shape of objects is preserved during instance and semantic segmentation stages, while the content and texture are maintained throughout the mask map to image synthesis stage. However, with the model proposed by Liu et al. [13], the overall shape is highly preserved, but the content and texture are not rich since it is based on the coloring idea. On the other hand, EdgeGAN [16] produces good content/texture; however, since it converts the sketches to edges via common learned representations, the shapes of objects are not well-preserved. As for PITI [60], most of the time, the shape along with the texture are not well preserved. This may be attributed to the fact that the model is trained on only one type of edge map, namely, HED [39], and it generates images for the foreground and the background altogether in one-shot. The visual results generated by our method and baselines are shown in Fig. 7.

4.3 Qualitative results

To evaluate the generated images qualitatively, a perceptual study is carried out to assess the synthesized images. Two factors are evaluated. The first aspect is realism which evaluates the quality of the generated images. The second aspect is the image

Table 2 IS score [47] of our proposed method and previous methods, in particular, Liu et al. [13], EdgeGAN [16], and PITI [60]

Method	IS	
	Mean score ↑	Std ↑
EdgeGAN [16]	3.827	0.790
Liu et al. model [13]	4.388	0.783
PITI [60]	4.565	0.499
Ours (object-level)	4.9898	0.879
Ours (scene-level)	**5.168**	**1.047**

Bold font indicates best result obtained for image synthesis in terms of IS score [47]

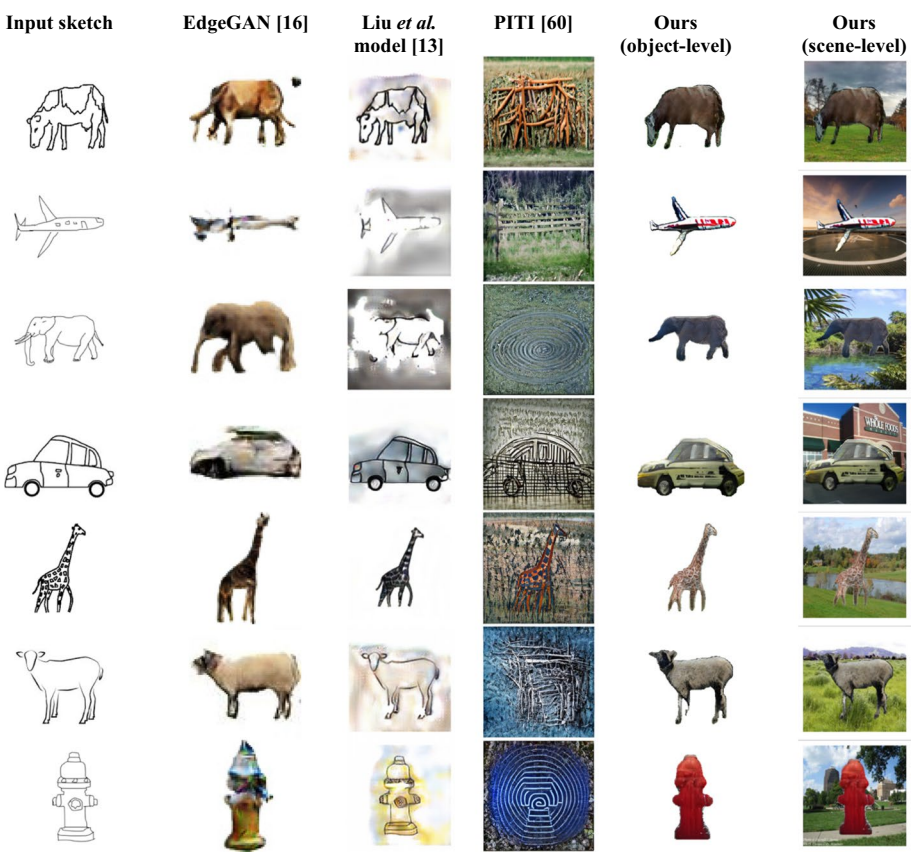
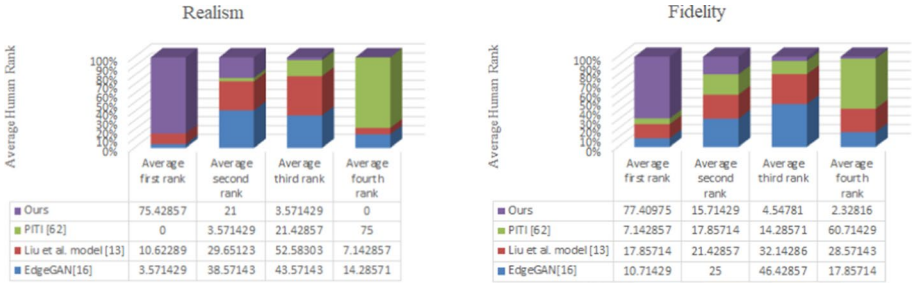


Fig. 7 Visualization of the results generated by our method and baselines [13, 16, 60]. Our results are with high quality and relevant background

fidelity, how similar the generated images are to their input sketches. 35 participants who aged between 20 and 50 years old are asked to rank the synthetic images, generated by our method and the baselines, in terms of the most realistic image on a scale of 1 to 4, where 1 means most realistic and 4 means less realistic. Object(s) label are given for each sample. Moreover, they are requested to rank the resulting images based on the similarity to the input sketch and the coloring quality with the same scale criteria, where 1 means most similar and 4 means less similar to the input sketch. Then, Average Human Rank (*HR*) is computed based on the user ranking choices and summarized in Fig. 8. As shown in this figure, our proposed method outperforms the baselines in both photo realism and image fidelity factors by a significant margin, where it achieves 75.43 and 77.41 for realism and fidelity, respectively as the first best rank among other generated images generated by the baselines, meaning that our method reflects more realism and better faithfulness. In the meantime, as for realism first rank, EdgeGAN [16], Liu et al. model [13], and PITI [60] reach only 3.57, 10.62, and 0, indicating less realism and quality. In addition, regarding fidelity factor first rank, EdgeGAN [16], Liu et al. model [13], and PITI [60] reach 10.71, 17.86, and 7.14, denoting less similarity to the corresponding input sketches. With regard to the worst



(a) Photo realism evaluation **(b) Image fidelity evaluation**

Fig. 8 A summary of Average Human Rank (HR) on the user ranking choices of our proposed method and the baselines [13, 16, 60]. Participants are requested to rank the generated images in terms of fidelity, the similarity to the input sketches, and the realism. Then, Average HR is computed and illustrated in this figure

case, our proposed framework is barely diagnosed as worst comparing to the baselines, where it obtains 0 and only 2.33 for realism and fidelity, respectively.

4.4 Ablation study

To analyze the efficacy of some components in our approach, we carry out an ablation study on MS-COCO dataset [37], where 5000 images are tested. We analyze the synthesized image after the image synthesis stage (*ISS*), after the background improvement stage (BIS), and after the face reconstruction stage (FRS). The reason is to specify whether the post-processing steps are working better and enhancing the synthesis results. FID [46] and IS [47] are computed after each of the mentioned stages. Moreover, various instance segmentation models are used in our framework, in particular, Mask R-CNN [17], Cascade RCNN [18], HTC [19], QueryInst [20], and DetectoRS [21] to determine which model works the best; and thus, the best model is incorporated in our framework. Only dodging and burning [38] edge maps that obtains pencil sketch is used in the ablation study. The analysis is summarized in Table 3. As can be seen, our assumption of implementing a post-process step of improving the background and reconstructing the human face is significantly enhance the generated images in all different instance segmentation models. Furthermore, DetectoRS model [21] achieves the best results with lowest FID score [46] and highest IS score [47] with 42.016 and 18.656, respectively. The example pencil sketches, and their corresponding synthetic images generated by our model are shown in Fig. 9. To aid our method to generalize better on freehand testing sketches of different types, we further train DetectoRS [21] on several types of edge maps, namely, dodging and burning [38], Holistically-nested edge detection [39], Canny edge detector [40], and Sobel operators [41].

5 Analysis and discussion

In the digital world and daily life, images are considered as one of the significant elements that can abbreviate a long story. A single image can tell a long story/fact, while a single word cannot. Further, a single image can convey multiple meanings based on the viewers' point of view, while usually a single sentence can only maintain one meaning. Thus, images are considered an important source of information and knowledge.

Table 3 The ablation study of our framework with different instance segmentation models.

Instance segmentation model	Component	FID [46] ↓	IS [47]	
			Mean score ↑	Std ↑
Mask R-CNN [17]	ISS	81.279	13.539	0.557
	BIS	46.234	16.295	0.492
	FRS	45.170	17.328	0.502
Cascade RCNN [18]	ISS	82.754	13.449	0.579
	BIS	46.208	16.406	0.549
	FRS	45.096	17.427	0.577
HTC [19]	ISS	73.146	14.853	0.342
	BIS	43.159	17.092	0.689
	FRS	42.019	18.152	0.686
QueryInst [20]	ISS	85.314	13.924	0.854
	BIS	45.964	17.054	0.680
	FRS	44.927	18.083	0.722
DetectoRS [21]	ISS	75.090	14.659	0.638
	BIS	43.129	17.590	0.734
	FRS	**42.016**	**18.656**	**0.739**

Bold font indicates best result obtained for image synthesis in terms of FID [46] and IS score [47]

However, creating an image is not only time-consuming, but also a tedious and painful task. Additionally, it requires artistic skills or expertise in the field or the software. Hence, creating an image from scratch is not a trivial task since it contains rich

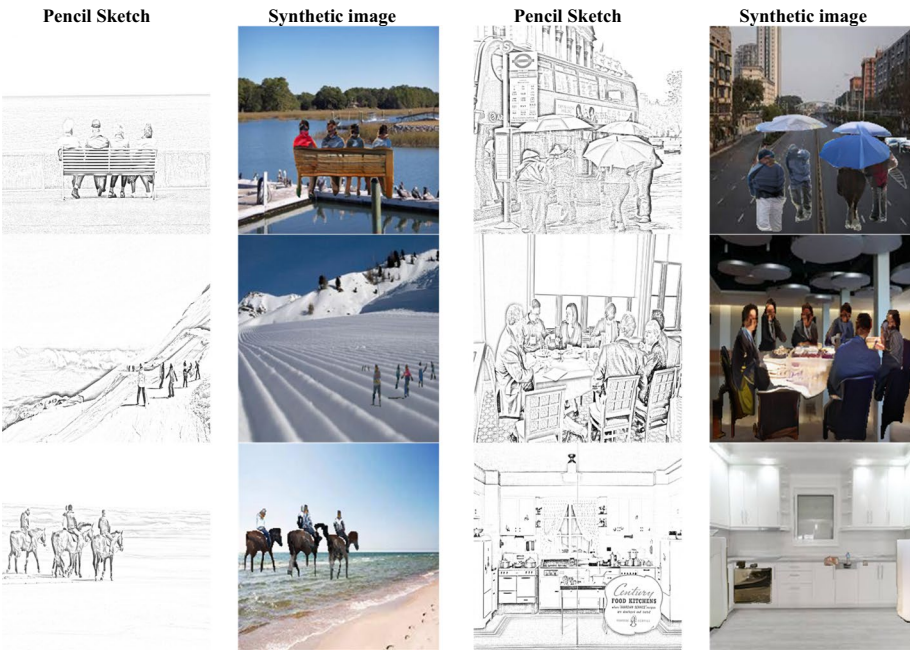


Fig. 9 Visualization of pencil sketches and their corresponding synthetic images generated by our model

features and fine-grained details such as colors, brightness, saturation, luminance, texture, shadow, just to name a few. To overcome this, an image synthesis, in particular, a sketch-to-image synthesis is the best approach to generate a colored image in no time through only a simple and rough black and white sketch. In fact, sketch-to-image synthesis reduces the burden of creating an image from scratch since it only requires a black and white rough and simple sketch with key structural information. Then, the input sketch is automatically mapped without human interventions into the corresponding image. Therefore, anyone can create an image even without artistic skills or expertise in the field or the software and in no time. This aids artists, photographers, and animation makers to reduce the repetitive work.

Since sketches are usually abstract and imperfect and mostly contain only the key structural information, converting a sketch into a colored image is a very challenging task that recently attracts the researchers' attention. Many studies have been conducted in this field, but the results are still not natural and realistic, especially with complex scenes with multiple objects. The reason is that the image is synthesized from the input sketch directly end-to-end in one shot, leading to an unnaturalistic and unsatisfactory image.

To this end, the key innovation and novelty of our proposed framework is decomposing the sketch-to-image synthesis problem into two sub-problems. Rather than generating an image directly from the input sketch, as most of the state-of-the-art sketch-to-image synthesis methods follow, we propose to first generate an intermediate result, e.g., semantic mask map, from the input sketch. Then, the color image is generated from the semantic mask map. This helps in generating more realistic image than the previous work. The reason is that our proposed framework first concentrates on generating similar objects' shape in the first intermediate result through the instance and semantic segmentation stages. Next, it concentrates on the content/texture of the objects and the fine-grained details via semantic mask to image synthesis stage. Therefore, our framework concentrates on one factor at a time, i.e., fidelity and then realism and quality, leading to better results. Moreover, our proposed method is performed in two levels: background segmentation and foreground segmentation so that no conflicts or inconsistency occurs. Furthermore, since our method is trained on different edge maps, it is able to generate images from various sketch styles. However, other state-of-the-art sketch-to-image models generate images directly in one-shot, focusing on both shape and texture at once, leading to unnaturalistic images. In addition, most of the previous methods are either ignoring the background generation or generating the background at the time of generating the foreground, leading to inconsistency. Indeed, one important reason of the deficiency in the existing approaches is that models are trained on only one style of sketches or one type of edge maps. Thus, they are unable to generate good results when the input sketches are different than what they trained on.

6 Conclusion and future work

In this paper, we propose a novel approach for sketch-to-image synthesis. Instead of generating the image directly from the sketch, we propose an intermediate output, in particular, mask map due to its ability to preserve the object shape, orientation, occlusion, and size. Therefore, using instance and semantic segmentation, the semantic mask map is generated from the input sketch. Following this, the mask map is translated into an image. Our method shows a significant improvement and outperforms the state-of-the-art sketch-to-image synthesis methods. Moreover, since our method is trained on different types of edge

maps, it is able to synthesize an image from different types of sketches. Furthermore, our method is capable of generating images of complex scenes, such as MS-COCO dataset [37]. In the future, we plan to improve our framework further by approximating the input sketch to the nearest edge map while preserving the shapes. This could be accomplished by learning a common representation for each object via a cross-modality learning to transfer the knowledge between different modalities, i.e., sketches and edge maps. This would support to generate images from very abstract sketches.

Acknowledgements The first author would like to thank Umm Al-Qura University, in Saudi Arabia, for the continuous support. The second author is supported by NSF grant # 2025234. This work has been supported in part by the University of Dayton Office for Graduate Academic Affairs through the Graduate Student Summer Fellowship Program.

Data availability The datasets generated during and/or analyzed during the current study are available from the corresponding author on reasonable request.

Declarations

Conflict of interest The authors have no conflicts of interest to declare that are relevant to the content of this article.

References

1. Chen T, Cheng M-M, Tan P, Shamir A, Hu S-M (2009) Sketch2Photo: internet image montage. ACM Trans Graph 28(5):1–10
2. Eitz M, Richter R, Hildebrand K, Boubekeur T, Alexa M (2011) Photosketcher: interactive sketch-based image synthesis. IEEE Comput Graph Appl 31(6):56–66
3. Szántó B, Pozsegovics P, Vámossy Z, Sergyán S (2011) Sketch4match — Content-based image retrieval system using sketches. In: IEEE 9th International Symposium on Applied Machine Intelligence and Informatics (SAMI), Smolenice, pp 183–188. https://doi.org/10.1109/SAMI.2011.5738872
4. Rajput GG, Prashantha (2019) Sketch based image retrieval using grid approach on large scale database. Procedia Comput Sci 165:216–223
5. Springenberg JT , Dosovitskiy A, Brox T, Riedmiller MA (2014) Striving for simplicity: the all convolutional net. In: International Conference on Learning Representations
6. Yu Q, Liu F, Song Y-Z, Xiang T, Hospedales TM, Loy CC (2016) Sketch me that shoe. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp 799–807
7. Sangkloy P, Burnell N, Ham C, Hays J (2016) The sketchy database: learning to retrieve badly drawn bunnies. ACM Trans Graph 35(4):1–12
8. Goodfellow IJ, Pouget-Abadie J, Mirza M, Xu B, Warde-Farley D, Ozair S, Courville A, Bengio Y (2014) Generative adversarial nets. In: Proceedings of the 27th International Conference on Neural Information Processing Systems 2:2672–2680
9. Sangkloy P, Lu J, Fang C, Yu F, Hays J (2017) Scribbler: controlling deep image synthesis with sketch and color. In: IEEE conference on computer vision and pattern recognition (CVPR), pp 6836–6845
10. Liu Y, Qin Z, Luo Z, Wang H (2017) Auto-painter: cartoon image generation from sketch by using conditional generative adversarial networks. ArXiv, abs/1705.01908
11. Xian W, Sangkloy P, Agrawal V, Raj A, Lu J, Fang C, Yu F, Hays J (2017) TextureGAN: controlling deep image synthesis with texture patches. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2018:8456–8465
12. Chen W, Hays J (2018) SketchyGAN: towards diverse and realistic sketch to image synthesis. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp 9416–9425
13. Liu R, Yu Q, Yu SX (2020) Unsupervised sketch to photo synthesis. Computer Vision – ECCV 2020. Springer International Publishing, Cham, pp 36–52
14. Liu B, Zhu Y, Song K, Elgammal A (2021) Self-supervised sketch-to-image synthesis. Proc Conf AAAI Artif Intell 35(3):2073–2081

15. Zhang P, Zhang B, Chen D, Yuan L, Wen F (2020) Cross-domain correspondence learning for exemplar-based image translation. In: IEEE/CVF conference on computer vision and pattern recognition (CVPR), pp 5142–5152
16. Gao C, Liu Q, Xu Q, Wang L, Liu J, Zou C (2020) SketchyCOCO: image generation from freehand scene sketches. In: Conference on Computer Vision and Pattern Recognition (CVPR), pp 5173–5182
17. He K, Gkioxari G, Dollar P, Girshick R (2017) Mask R-CNN. In: IEEE international conference on computer vision (ICCV), pp 2961–2969
18. Cai Z, Vasconcelos N (2017) Cascade R-CNN: Delving into high quality object detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 6154–6162
19. Chen K, Pang J, Wang J, Xiong Y, Li X, Sun S, Feng W, Liu Z, Shi J, Ouyang W, Loy CC, Lin D (2019) Hybrid Task Cascade for Instance Segmentation. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 4974–4983
20. Fang Y, Yang S, Wang X, Li Y, Fang C, Shan Y, Feng B, Liu W (2021) Instances as Queries. In: Proceedings of the IEEE/CVF international conference on computer vision, pp 6910–6919
21. Qiao S, Chen L-C, Yuille A (2021) DetectoRS: detecting objects with recursive feature pyramid and switchable atrous convolution. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 10213–10224
22. Park T, Liu M-Y, Wang T-C, Zhu J-Y (2019) Semantic image synthesis with spatially-adaptive normalization. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 2337–2346
23. Eitz M, Hildebrand K, Boubekeur T, Alexa M (2009) A descriptor for large scale image retrieval based on sketched feature lines. In: Proceedings of the 6th Eurographics symposium on sketch-based interfaces and modeling. pp 29–36
24. Manjunath BS, Salembier P, Sikora T (eds) (2002) Introduction to MPEG-7: Multimedia Content Description Interface, Chichester, England, John Wiley & Sons
25. Chalechale A, Mertins A, Naghdy G (2004) Edge image description using angular radial partitioning. IEE Proc Vis Image Signal Process 151(2):93
26. Dalal N, Triggs B (2005) Histograms of oriented gradients for human detection. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) 1:886–893
27. Lowe DG (2004) Distinctive image features from scale-invariant keypoints. Int J Comput Vis 60(2):91–110
28. Ojala T, Pietikäinen M, Harwood D (1996) A comparative study of texture measures with classification based on featured distributions. Pattern Recogn 29(1):51–59
29. Kruskal JB (1964) Nonmetric multidimensional scaling: a numerical method. Psychometrika 29(2):115–129
30. Krause A "A classification based similarity metric for 3D image retrieval," Cmu.edu, 01-Jun-1998. [Online]. Available: https://www.ri.cmu.edu/publications/a-classification-based-similarity-metric-for-3d-image-retrieval/. [Accessed: 20-Sep-2021]
31. Chicco D (2021) Siamese neural networks: An overview. Methods Mol Biol 2190:73–94
32. Yu Q, Yang Y, Liu F, Song Y-Z, Xiang T, Hospedales TM (2017) Sketch-a-net: A deep neural network that beats humans. Int J Comput Vis 122(3):411–425
33. Krizhevsky A, Sutskever I, Hinton GE (2017) ImageNet classification with deep convolutional neural networks. Commun ACM 60(6):84–90
34. Szegedy C, Liu W, Jia Y, Sermanet P, Reed S, Anguelov D, Erhan D, Vanhoucke V, Rabinovich A (2015) Going deeper with convolutions. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 1–9
35. Zou C, Mo H, Gao C, Du R, Fu H (2019) Language-based colorization of scene sketches. ACM Trans Graph 38(6):1–16
36. Isola P, Zhu JY, Zhou T, Efros AA (2017) Image-to-image translation with conditional adversarial networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 1125–1134
37. Lin TY, Maire M, Belongie S, Hays J, Perona P, Ramanan D, Dollár P, Zitnick CL (2014) Microsoft coco: Common objects in context. In: Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, Proceedings, Part V 13 2014, pp 740–755
38. Beyeler M (2015) OpenCV with Python blueprints: design and develop advanced computer vision projects using OpenCV with Python. Packt Publishing Ltd., London, England, ISBN 978-178528269-0,
39. Xie S, Tu Z (2015) Holistically-nested edge detection. In: Proceedings of the IEEE International Conference on Computer Vision (ICCV), pp 1395–1403
40. Ding L, Goshtasby A (2001) On the canny edge detector. Pattern Recogn 34(3):721–725

41. Kanopoulos N, Vasanthavada N, Baker RL (1988) Design of an image edge detection filter using the Sobel operator. IEEE J Solid State Circuits 23(2):358–367
42. Caesar H, Uijlings J, Ferrari V (2018) COCO-stuff: thing and stuff classes in context. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp 1209–1218
43. Zhou B, Lapedriza A, Khosla A, Oliva A, Torralba A (2018) Places: a 10 million image database for scene recognition. IEEE Trans Pattern Anal Mach Intell 40(6):1452–1464
44. Soo S (2014) Object detection using Haar-cascade classifier. Institute of Computer Science, University of Tartu 2(3):1–2
45. Wan Z, Zhang J, Chen D, Liao J (2021) High-fidelity pluralistic image completion with transformers. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp 4692–4701
46. Heusel M, Ramsauer H, Unterthiner T, Nessler B, Hochreiter S (2017) "Gans trained by a two time-scale update rule converge to a local Nash equilibrium," in NIPS
47. Salimans T, Goodfellow I, Zaremba W, Cheung V, Radford A, Chen X (2016) Improved techniques for training GANs. In: Proceedings of the 30th International Conference on Neural Information Processing Systems, pp 2234–2242
48. Dokmanic I, Parhizkar R, Ranieri J, Vetterli M (2015) Euclidean distance matrices: essential theory, algorithms and applications. Adv Neural Inf Process Syst 2016;29
49. Wang L, Qian X, Zhang Y, Shen J, Cao X (2020) Enhancing sketch-based image retrieval by CNN semantic re-ranking. IEEE Trans Cybern 50(7):3330–3342
50. Li Z, Deng C, Yang E, Tao D (2021) Staged sketch-to-image synthesis via semi-supervised generative adversarial networks. IEEE Trans Multimedia 23:2694–2705. https://doi.org/10.1109/TMM.2020.3015015
51. Mirza M, Osindero S (2014) Conditional generative adversarial nets. arXiv [cs.LG]
52. Osahor U, Kazemi H, Dabouei A, Nasrabadi N (2020) Quality guided sketch-to-photo image synthesis. In 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW, 2020)
53. Liu B, Zhu Y, Song K, Elgammal A (2020) Self-supervised sketch-to- image synthesis. In: Proceedings of the AAAI conference on artificial intelligence 2021 May 18, vol 35, no. 3, pp 2073–2081
54. Kramer MA (1991) Nonlinear principal component analysis using autoassociative neural networks. AIChE J 37(2):233–243
55. Vincent P, Larochelle H, Lajoie I, Bengio Y, Manzagol PA, Bottou L (2010) Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. Journal of machine learning research. 11(12)
56. Feng Z, Xu C, Tao D (2019) Self-supervised representation learning by rotation feature decoupling. In 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR, page 10364–10374
57. Kolesnikov A, Zhai X, Beyer L (2019) Revisiting self-supervised visual representation learning. arXiv [cs.CV]
58. Liu L, Chen R, Wolf L, Cohen-Or D (2010) Optimizing photo composition. Comput Graph. Forum 29(2):469–478
59. Sohl-Dickstein J, Weiss EA, Maheswaranathan N, Ganguli S (2015) Deep unsupervised learning using nonequilibrium thermodynamics. In: International conference on machine learning. PMLR, pp 2256–2265
60. Wang T, Zhang T, Zhang B, Ouyang H, Chen D, Chen Q, Wen F (2022) "Pretraining is all you need for image-to-image translation," arXiv [cs.CV]
61. Ho J, Jain A, Abbeel P (2020) Denoising diffusion probabilistic models. Adv Neural Inf Process Syst 33:6840–6851
62. Yang S, Ermon S (2020) Improved techniques for training score- based generative models. Adv Neural Inf Process Syst 33:12438–12448
63. Jolicoeur-Martineau A, Piché-Taillefer R, Combes RTD, Mitliagkas I (2020) Adversarial score matching and improved sampling for image generation. arXiv preprint arXiv:2009.05475
64. Nichol A, Dhariwal P (2021) Improved denoising diffusion probabilistic models. In: International Conference on Machine Learning. PMLR, pp 8162–8171
65. Sasaki H, Willcocks CG, Breckon TP (2021) Unit-ddpm: Unpaired image translation with denoising diffusion probabilistic models. arXiv preprint arXiv:2104.05358
66. Özbey M, Dalmaz O, Dar SUH, Bedel HA, Özturk Ş, Güngör A, Çukur T (2022) Unsupervised medical image translation with adversarial diffusion models. IEEE Transactions on Medical Imaging
67. Güngör A, Dar SUH, Ztürk ŞÖ, Korkmaz Y, Elmas G, özbey M, Çukur T (2022) Adaptive diffusion priors for accelerated mri reconstruction. Medical Image Analysis. 2023 Jun 20:102872.

68. Nichol AQ, Dhariwal P, Ramesh A, Shyam P, Mishkin P, McGrew B, Sutskever I, Chen M (2022) GLIDE: towards photorealistic image generation and editing with text-guided diffusion models. arXiv preprint arXiv:2112.10741. 2021 Dec 20
69. Ho J, Saharia C, Chan W, Fleet DJ, Norouzi M, Salimans T (2022) Cascaded diffusion models for high fidelity image generation. The Journal of Machine Learning Research. 2022 Jan 1;23(1):2249-81.
70. Ramesh A, Dhariwal P, Nichol A, Chu C, Chen M (2022) "Hierarchical text-conditional image generation with clip latents," arXiv preprint arXiv:2204.06125
71. Xu P, Hospedales TM, Yin Q, Song Y-Z, Xiang T, Wang L (2020) "Deep learning for free-hand sketch: A survey," arXiv [cs.CV]
72. Qi Y, Su G, Wang Q, Yang J, Pang K, Song Y-Z (2022) Generative sketch healing. International Journal of Computer Vision. vol. 130, no. 8, pp. 2006–2021.
73. Sun J, Yu H, Zhang JJ, Dong J, Yu H, Zhong G (2022) Face image-sketch synthesis via generative adversarial fusion. Neural Networks. vol. 154, pp. 179–189.
74. Otsu N (1979) A threshold selection method from gray-level histograms. IEEE Trans Syst Man Cybern 9(1):62–66
75. Baraheem SS, Nguyen TV (2020) Text-to-image via mask anchor points. Pattern Recogn Lett 133:25–32
76. Baraheem SS, Nguyen TV (2020) "Aesthetic-aware text to image synthesis," in 2020 54th Annual Conference on Information Sciences and Systems (CISS), pp. 1–6
77. Richardson E et al. (2020) "Encoding in Style: a StyleGAN Encoder for Image-to-Image Translation," arXiv [cs.CV], pp. 2287–2296
78. Liu M, Li Q, Qin Z, Zhang G, Wan P, Zheng W (2021) BlendGAN: Implicitly GAN blending for arbitrary stylized face generation. Advances in Neural Information Processing Systems 34:29710–22
79. Li B, Zhu Y, Wang Y, Lin C-W, Ghanem B, Shen L (2021) AniGAN: Style-guided generative adversarial networks for unsupervised Anime face generation. IEEE Transactions on Multimedia 24:4077–91
80. Liu B, Song K, Elgammal A (2020) Sketch-to-art: Synthesizing stylized art images from sketches. Computer Vision – ACCV 2020, Cham: Springer International Publishing, 2021, pp. 207–222
81. Tian Y, Suzuki C, Clanuwat T, Bober-Irizar M, Lamb A, Kitamoto A (2020) KaoKore: A Pre-modern Japanese Art Facial Expression Dataset. arXiv preprint arXiv:2002.08595
82. Chen Z, Chen L, Zhao Z, Wang Y (2020) AI illustrator: Art illustration generation based on generative adversarial network. IEEE 5th International Conference on Image, Vision and Computing (ICIVC), 2020, pp. 155–159
83. Tian Q, Franchitti J-C (2022) Text to artistic image generation. arXiv preprint arXiv:2205.02439
84. Shen J, Robertson N (2021) BBAS: towards large scale effective ensemble adversarial attacks against deep neural network learning. Information Sciences. vol. 569, pp. 469–478.
85. Yang B, Zhang H, Zhang Y, Xu K, Wang J (2021) Adversarial example generation with AdaBelief Optimizer and Crop Invariance. Appl Intelligence 53(2):2332–47
86. Kwon H, Jeong J (2022) AdvU-net: generating adversarial example based on medical image and targeting U-net model. J Sens 2022:1–13
87. Iyyer M, Wieting J, Gimpel K, Zettlemoyer L (2018) Adversarial example generation with syntactically controlled paraphrase networks. arXiv preprint arXiv:1804.06059
88. Zhang R, Luo S, Pan L, Hao J, Zhang J (2022) Generating adversarial examples via enhancing latent spatial features of benign traffic and preserving malicious functions. Neurocomputing 490:413–430