Geometric Deformation on Objects: Unsupervised Image Manipulation via Conjugation *

Changqing Fu[1] and Laurent Cohen[1]

CEREMADE, UMR CNRS 7534,
Universit Paris Dauphine, PSL
Place du Marechal de Lattre de Tassigny
75775 Paris cedex 16, France
{cfu,cohen}@ceremade.dauphine.fr
https://www.ceremade.dauphine.fr

Abstract. A novel two-stage approach is proposed for image manipulation and generation. User-interactive image deformation is performed through editing of contours. This is performed in the latent edge space with both color and gradient information. The output of editing is then fed into a multi-scale representation of the image to recover quality output. The model is flexible in terms of transferability and training efficiency.

Keywords: Machine Learning, Image Generation, Generative Adversarial Network, Image deformation, Contour Editing

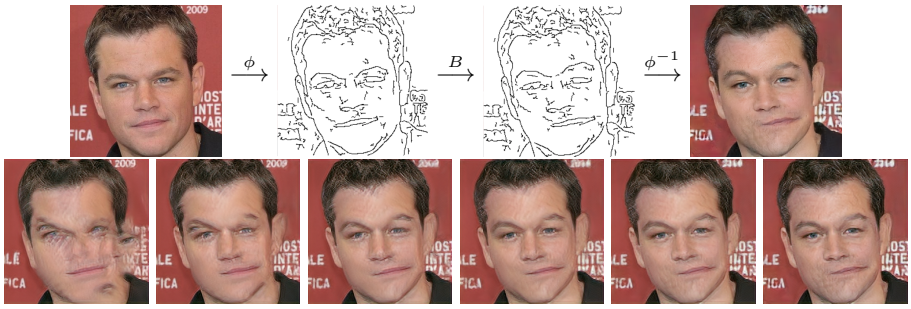

Fig. 1. Top row, from left to right: a) Input. b) Reconstruction from sparse contour. c) Reconstruction from deformed contour, moving the lip and eyebrow contours via user interaction. Bottom row, from left to right: Multi-scale Super-resolution of the deformed Reconstruction.

* This work was funded in part by the French government under management of Agence Nationale de la Recherche as part of the "Investissements davenir" program, reference ANR-19-P3IA-0001 (PRAIRIE 3IA Institute).

1 Introduction

Image manipulation task is among one of the fast-growing fields in Computer Vision. Many existing algorithms for image editing are supervised and not adaptive to new data, and often require fine-tuning and slow training to achieve desirable performance. Therefore, the motivation of this work is to propose an alternative unsupervised way to refine the reconstruction, providing flexibility for difficult training and limited dataset.

In our setting of image manipulation, we introduce two manifolds $\mathcal{M}$ and $\mathcal{N}$, where $\mathcal{M}$ denotes the space of natural images, and $\mathcal{N}$ is the space of contour representation of the images. Function A in $\mathcal{M}$ stands for the desired final editing effect, whereas B in $\mathcal{N}$ is user's editing in the latent space. Since A is often complicated to implement, we alternatively perform editing B in the hidden contour space $\mathcal{N}$ via a pull-back mapping.

Mathematically speaking, Conjugation, or Similarity Transformation, refers to a pair of transformations $A : \mathcal{M} \longrightarrow \mathcal{M}$ and $B : \mathcal{N} \longrightarrow \mathcal{N}$ that satisfies $A = \phi^{-1} \circ B \circ \phi$, where ϕ is a diffeomorphism between two manifolds $\mathcal{M}$ and $\mathcal{N}$. In our case, ϕ is a well-defined image contour detector. The invertibility of ϕ is necessary for the well-definedness of the conjugation, but it is an ill-posed inverse problem. With the recent advancement of CNN-based Image Translation models, the inversion map ϕ^{-1} can be learned in a data-driven manner.

One of the advantages of contour representation is its sparsity, since any moving or distorting operation will create no effect on flat or zero-valued areas, but will create discontinuity effect on a natural image. Moving or distorting operation can be easily manipulated by human interaction on a computer, and thus is practically meaningful. Furthermore, contour information is not enough to recover an image, and therefore color and gradient information are helpful to improve the contour's representation ability without adding to sparsity.

On top of the conjugation paradigm, a strategy similar to deep internal learning[23] is performed. Instead of storing all the information in a single model to produce a high-quality image, we make use of the input image itself to carve details on the output image. This method is especially good at producing textures similar to the input image, even if they are never seen in the training database.

The organization of this paper is as follows. After reviewing related work in Section 2, we will present the main ideas of our work in Section 3, including sparse contour and multi-scale representations, and then detail the algorithms involved in Section 4, illustrated by examples on various databases.

Our contribution is that a natural Downscaling-Reconstruction strategy is proposed as a post-processing approach to obtain high-fidelity output for image manipulation, and it requires very short training time, and at the same time provides better transferability comparing to the existing approach.

2 Related Works

There are lines of research that understand the latent space $\mathcal{M}$ as sparse representation [4], color representation [28], or both [19]. Recently there are works

which deals with implicit hidden representation using inner features [7][8], providing the model with explainability. In this paper, we focus on geometric manipulation in the hidden space, and especially recovery in a new perspective under both supervised and unsupervised setting.

Sketch translation [3] [14] aims to produce realistic images out of abstract sketches drawn by human. Natural images can be produced even with messy or cartoon-like input. Therefore, the precision of geometric constraints is compromised on as a systematic side effect of the trade-off. These methods are to tackle challenges like Sketchy [18], QuickDraw [10], or ShoeV2/ChairV2 datasets [27], whereas the contours in our method is similar to Edge2shoes[26] or Edges2handbags [28], according to their realistic structure. Therefore, our focus is towards image synthesis, rather than image retrieval in the database or on the image manifold. It does not require any a priori edge information, since this is already incorporated inside the image itself, and could be computed using efficient context-aware edge detectors (section 3.1).

On the other hand, the multi-scale image structure is motivated by SinGAN[21], a multiscale invariant of InGAN[22]. Other super-resolution techniques such as [25] [6] exist but only tackle the standard super-resolution problem. The hierarchical structure of SinGAN enables more flexible input, and is able to recover not only inputs of low-resolution, but also color shapes, even those created from scrach by human.

3 Image Model

The intuition of our approach is twofold, consisting of two different ways of understanding for neural image perception.

3.1 Sparse Contour Representation

Fig. 2. Recovery from Sparse Representation. From left to right: Real image, Contour representation, Low frequency reconstruction, High frequency reconstruction.

Given image space $\mathcal{M}$ and contour space $\mathcal{N}$, in order to achieve a robust diffeomorphism $\phi : \mathcal{M} \longrightarrow \mathcal{N}$ between the edge representation space and the image space, as is mentioned, the invertibility of ϕ is the key to the recovery result. More precisely, ϕ^{-1} is naturally defined as $\phi^{-1} : \mathcal{N} \longrightarrow 2^{\mathcal{M}}$, $\phi^{-1}(y) = \{x \in$

$\mathcal{M}|\phi(x) = y\}, \forall y \in \mathcal{N}$, and the pre-image of contour is not unique. Minimizing the GAN energy (Equation 1) further finds the point in the pre-image which looks like the image database the most. This result in $\widehat{\phi^{-1}} : \mathcal{N} \longrightarrow \mathcal{M}$, a well-defined surrogate in a data-driven sense.

In practice, we use an edge detector as ϕ, and apply a Generator network to fit the function ϕ^{-1}, together with its Discriminator counterpart. This is under the scope of Image Translation problem.

In general, image translation problem is described as follows. Let $\mathcal{X} = \mathcal{Y} = \mathbb{R}^{3\times H\times W}$ be two image spaces of fixed size $H \times W$, with training examples $x_i \in \mathcal{X}, y_i \in \mathcal{Y}, i = 1, \ldots, N$. Supervised Learning approaches for image translation aims to learn a map $\widehat{\phi^{-1}}$ from $\mathcal{X}$ to $\mathcal{Y}$ by minimizing the loss function $\mathcal{L}(\phi(x_i), y_i)$. The fitted function is then used to generate images following the same distribution in $\mathcal{M}$. The restriction of this map on the sketch manifold $G := \widehat{\phi^{-1}_{|\mathcal{N}}} : \mathcal{N} \subseteq \mathcal{X} \longrightarrow \mathcal{M}$ is supposed to be an onto map to the image manifold $\mathcal{M} \subseteq \mathcal{Y}$, which is often called a Conditional Generator since it has a complex prior distribution on $\mathcal{X}$, and is obtained by minimizing the Adversarial Loss in equation (1).

Now we discuss some properties of G:

- Ideally, the onto property of G on $\mathcal{M}$ is often called generalization ability, whereas the lack of onto property on $\mathcal{M}$ and the fact that the image of G is restricted on $\{y_i\}_{i=1}^N \subseteq \mathcal{N}$ is called overfitting. Moreover, the lack of onto property of G on $\{y_i\}_{i=1}^N \subseteq \mathcal{N}$ is referred to as Mode Collapse.
- In practice, for the empirical $\widehat{\phi^{-1}}$, the pre-image of $\mathcal{M}$ is not necessarily equal to $\mathcal{N}$. In fact $\phi^{-1}(\mathcal{M}) \supseteq \mathcal{N}$. In other words, the map $\widehat{\phi^{-1}}^{-1}$ is not L^1-continuous, since the inverse of open ball in $\mathcal{M}$ is not open in $\mathcal{N}$ under L^1 topology. Illustrations are as follows in figure (3). We perform a differential attack or latent recovery [24] on a pretrained Pix2Pix [12] model, resulting in a non-sparse noisy pre-image. Instead of recovering the latent code as white noise in the original work, our input has meaningful structure. The fact that c) is not a clean sketch shows that for L^1 neighbours in $\mathcal{M}$, a) and d), their pre-image are not neighbours in $\mathcal{N}$, implying that $\widehat{\phi^{-1}}^{-1}$ is far from smooth. Therefore $\widehat{\phi^{-1}}$ is not an L^1-diffeomorphism itself. However, by restricting $\widehat{\phi^{-1}}$ on the contour manifold $\mathcal{N}$, the resulting $G = \widehat{\phi^{-1}_{|\mathcal{N}}}$ turns out to be a proper counterpart for ϕ to recover image from geometrical constraints.

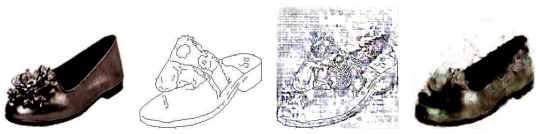

Fig. 3. From left to right: a) Original Image, b) Initialization for the Pre-Image Search, c) Pre-image Found, d) Reconstruction from c)

The edge-detector/translator pair ϕ and G between $\mathcal{M}$ and $\mathcal{N}$ are detailed as follows:

Edge Detector A tree-based edge detection algorithm [5] is applied as edge extractor ϕ. The random forest is trained with samples from BSDS500 dataset with structured labels of edge and segmentation. Hard thresholding is then applied with a pre-defined threshold to produce an edge mask. We follow [4] to use an N-channel contour representation but with modifications. **a)** Different from their fixed sparsity rate, we used a fixed threshold since our image database naturally have contours of diverse sparsity. **b)** We extract both 3D-color and 6D-gradient information on the contour pixels without computing that of both side of the contour (proposed in the original paper). This solution reduces computation and at the same time avoids contour overlapping after user's editting to the greatest extend.

Image Generator A U-Net[17] is applied as the baseline algorithm to approximate ϕ^{-1}. [4] refer to this model as the Low Frequency Network (LFN), which produces an intermediate output, and propose to apply another U-Net on top of the Low Frequency output and the contour input. The second network is called the High Frequency Network (HFN) since it produces finer details. It's worth noting that: **a)** During the optimization of the HFN, it's optional to update the weights of LFN in the meantime, since the computational graph of back-propagation can reach layers in the first network. This optional operation will distort the output of LFN, since the training process for the second network does not anymore aim to minimize the L^1 distance between the Low Frequency output and the training target. **b)** The second network adds complexity to the model, but it does not provide additional information to the model input. In fact, the concatenation operation in the U-Net plays a similar role, since it is nothing but concatenating inputs along with the intermediate outputs. HFN and concatenation operations are meaningful since they enforce the generator with intermediate layer information, and the secondary goal, low frequency network, is both meaningful and explainable. A recent work [15] is similar to this idea, and it inspires us that both networks could be trained simultaneously, encouraging LFN to be of even less frequency to be suitable for the input of the multi-scale model. In fact, flat and "quantized" input proves to be more effective in some cases[21]. **c)** Existing work does not tackle the cross-domain challenge, which motivates us to consider using the information outside the training database, namely the testing target itself.

With these considerations, we introduce the post-processing part of the model, based on every single test image.

3.2 Multi-scale Representation

Fig. 4. Optimal Reconstruction of an Image from coarse-grained scale.

We first introduce a sequence of RGB image space $V_n \subseteq \mathcal{X} = \mathbb{R}^{3\times H_n \times W_n}$, $n \in \mathbb{Z}$, for images of height H_n and width W_n monotonically increasing with respect to n. The coarse-grained space V_0 is of some fixed size $H_0 \times W_0$, and $x_N \in V_N$ denotes an image with camera resolution $H_N \times W_N$. $V_N = \mathcal{M}$ is the output space, which is the previously defined image manifold. By definition, $V_{-\infty} := \lim_{n\to-\infty} V_n$ is a single RGB pixel, whereas $V_\infty := \lim_{n\to\infty} V_n$ is the perfect vision with infinite resolution. For each n, down-scaling $\pi : x_n \in V_n \longmapsto x_{n-1} \in V_{n-1}$ from $x_n \in V_n$ with scaling factor $r \in (0, 1)$, since the corresponding upsampling map π^{-1} is a bijection between V_{n-1} and a linear subspace of V_n.

An image is represented by a Convolutional Neural Network. In other words, information in the image is memorized in weights of the neurons. More precisely, this is done by fitting a map G from randomly sampled noise to image data. An image generator is defined by $G : \mathcal{Z} \longrightarrow \mathcal{M} : z \longmapsto x_N$ and is trained to represent the image from the multi-scale code $z \in \mathcal{Z} = \prod_{n=1}^{N} \mathcal{Z}_n$, where $\mathcal{Z}_n = \mathbb{R}^d$ is the perturbation at each scale, for $n = 0, \ldots, N$. Mathematically, a map $G_\sharp$ between two probability distribution spaces is induced by the neural network G, where a smooth distribution in $\mathbb{R}^d$ is mapped to a probability distribution on the Image Manifold. This can be an empirical distribution, in the case of a database, or a Dirac distribution, in the case of a single image. A model that learns only to represent a single target is often referred to as Mode Collapse, which often causes low representation ability, signifying improper optimization. However, in our Super Resolution setting, our goal is to add texture details to the blurry input that are *unique* to the image. Therefore the flexibility over local perturbations on the input is the key to the model.

The unconditional GAN is defined as follows in an iterative form:

$$\begin{array}{ll} \text{Refinement:} & x_n = \widetilde{x_{n-1}} + G_n(\widetilde{x_{n-1}} + z_n), \quad n = 0, \ldots, N \\ \text{Upsampling:} & \widetilde{x_n} = \pi(x_n) \\ \text{Initialization:} & x_0 = G_0(z_0) \end{array}$$

where at scale n, x_n is the downsampled image at rate r^{N-n}, and z_n is white noise. As a result, the final Generator is given by

$$
\begin{aligned}
G(z) &:= G_N(z_0, \ldots, z_N) \\
&= \underbrace{\underbrace{G_0(z_0)}_{\triangleq x_0} + G_1(G_0(z_0) + z_1) + \cdots}_{\triangleq x_1} \\
&\quad + G_N(G_{N-1}(\underbrace{\cdots G_1(G_0(z_0) + z_1) \cdots}_{\triangleq x_{N-2}} + z_{N-1}) + z_N).
\end{aligned}
$$

Upsampling is omitted here for simplicity, by assuming that every object lives in V_∞.

Note that $r = \frac{1}{2}$ is the special case. When $H_{n-1} = \frac{1}{2}H_n, W_{n-1} = \frac{1}{2}W_n$, suppose $x_N \in V_N$ is an image, and suppose the corresponding 2D Haar wavelet decomposition is $x_N = \sum_{n=0}^{N} a_n \varphi_n$, then $\{\varphi_n\}_{n=1}^{N}$ are *orthogonal* and $a_n \varphi_n \in V_n$,

However, in practice, $r = \frac{1}{2}$ does not produce high-fidelity recovery, and $r = 3/4$ is a good balance between model complexity and quality. Upsampling operation π is performed by spline interpolation. z_n is chosen to be $\mathcal{N}(0, \sigma_n)$ with $\sigma_n \propto \|\pi(x_{n-1}) - x_n\|$ in order to match the intensity of randomness at each scale. In reconstruction/super-resolution task, perturbation z_n can be set to zero. The input image can be fed into any scale by down-scaling/up-scaling to obtain output of size greater than the input image, which adds to the detail of the image and achieves super-resolutions.

4 Algorithm

The minimization formulation for the multi-scale reconstruction problem is adapted from [12]. For each scale $n = 1, \ldots, N$,

$$
\min_G \max_D \mathcal{L} = \mathcal{L}_{adv}(G_n, D_n) + \alpha \mathcal{L}_{rec}(G_n) \tag{1}
$$

The **Adversarial Loss** $L_{adv}(G_n, D_n)$ is adapted from WGAN-GP[9]:

$$
\mathcal{L}_{adv}(G_n, D_n) = \mathbb{E}_x[D_n(\mathbf{x_n})] + \mathbb{E}_z[-D_n(G_n(\mathbf{x_n} + \mathbf{z_n}))] \tag{2}
$$

where $\mathbf{x_n} := (x_0, \ldots, x_n)$ and $\mathbf{z_n} := (z_0, \ldots, z_n)$ are sub-scale images and noise respectively.

In the sparse recovery case, the setting is similar and $\mathbf{x_n}$ is replaced with the real images y, and the perturbed down-scaled image $\mathbf{x_n} + \mathbf{z_n}$ is replaced with contour representation x.

$$
\mathcal{L}_{adv}(G, D) = \mathbb{E}_y[D(y)] + \mathbb{E}_x[-D(G(x))] \tag{3}
$$

This optimization objective is similar to the Binary Cross-Entropy Loss in the Vanilla GAN:

$$\mathcal{L}_{adv}(G_n, D_n) = \mathbb{E}_x[\log D_n(x)] + \mathbb{E}_z[\log(1 - D_n(G_n(z)))] \tag{4}$$

Note that in Vanilla GAN, the discriminator $D : \mathcal{N} \longrightarrow [0, 1]$ could be understood as the probability that the input image is fake, and thus the adversarial loss could be treated as the log likelihood function. In comparison, our discriminator aims to detect fake images as positive or negative otherwise, as opposed to the original case.

Reconstruction Loss is given by

$$L_{rec}(G_n) = \|G_n(\pi(\widetilde{x_{n-1}}) + 0) - x_n\|^2. \tag{5}$$

This deterministic term ensures that each layer performs proper refinement and adds high-frequency information to the image.

4.1 Cross-domain Transferability

Fig. 5. Pretrained on VGG Face dataset. From left to right: a) Input. b) Reconstruction from supervised edge prior (Pix2Pix model trained on paired data). c) Reconstruction from detected edge (pretrained on Face Dataset). Original tensorflow implementation by [4]. d) Reconstruction from cleaned sparse edge. e) Our reconstruction result.

Figure 5 shows an example of cross-domain results of the model, using the original tensorflow implementation. The model was trained on the VGG Face Dataset[16], and tested on an image of a single object. Artefacts are produced with the cross-domain test sample, whereas our implementation trained on shoe dataset (right column) works well. However, this can be corrected by post-processing procedure (see Figure 7 below). We also show the baseline result of Pix2Pix on the second left column as comparison.

Fig. 6. Fast convergence for training –more contour reconstruction examples. From left to right: a) High Frequency Reconstruction b) Intermediate Low Frequency Reconstruction. c) Input Image. d) Sparse edge representation. This result can be obtained after *a few minutes* on an NVIDIA T4 GPU.

4.2 Contour Manipulation

We illustrate our results with contour translation (Figure 7) and contour removal (Figure 8) examples.

Fig. 7. From left to right: a) Input. b) Multi-scale Reconstruction. c),d) input and output of Pix2Pix model. e) Multi-scale Reconstruction of the Pix2Pix output. f) Reconstruction from contour representation, by manually moving the edge of the logo on the shoe. g) Multi-scale reconstruction of the deformation.

Figure 7 shows the robustness of the multi-scale post-processing in terms of transferability. The result is trained on face data and tested on shoe data. Even if the output of edge reconstruction has unexpected cool tone caused by cross-domain transfer learning (Second right in the figure), the information of the input image is still able to correct the details. The effectiveness of the post-processing for other tasks is presented as a bonus product (c, d, e in Figure 7), still not perfect but showing significant improvement over the previous outcome.

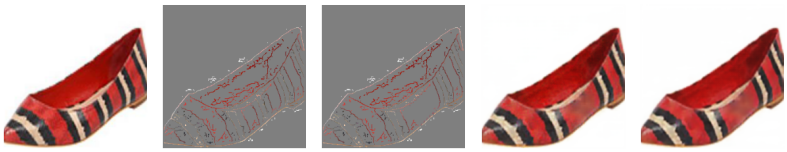

Fig. 8. From left to right: a) Input. b) Pre-image in parse edge space. c) Edge removal. d) Reconstruction of Low Frequency. d) Final Reconstruction.

Figure 8 shows the robustness of the sparse recovery method in terms of editing, quality and sparsity. Reconstruction is clean after a rough eraser edit. The figure is well recovered even from undertrained edge detector which produces noise on both sides of the contour. Recovery quality are not harmed by manually cleaning the noise on both sides.

Finally, figure 1 shows our final result where the supervised stage is trained independently on the VGG face dataset. The first row shows the validity of contour editing, and the second row presents the quality of post-processing. As can be seen, not only does the post-processing produce more natural skin color than the unprocessed reconstruction, it also adds to tiny randomness to the image so that the image is more diverse and privacy-protecting comparing to the input. The latter effect could be augmented by tuning α.

4.3 Implementation Details

Edge Detection Edge detection is performed[5] by training on a few samples from the BSDS500 dataset[1]. We find that this version of edge detector, though under-trained, well preserves color information for image recovery.

Network Structure For the Contour Reconstruction, we tested both U-net and ResNet Generators[29][13]. The U-net consists of Conv(3×3)-BatchNorm-LeakyReLU blocks of feature with concatenation operation. The ResNet[11] contains convolutions layers, several residual blocks, and then convolutions. For Multi-scale Reconstruction, we use ResNet of 5 convolution blocks of the form Conv(3×3)-BatchNorm-LeakyReLU[20]. The Discriminator are PatchGANs of fixed structure[12]. The number of patches depends on the input size. This is similar to a recent work[2] that propose to improve classification with Bag-of-words Patch features.

Training Strategy For Contour Reconstruction, we use the Adam optimizer with learning rate 0.001 with Cosine Annealing learning rate for 50 epochs on a mini dataset of 200 samples. For Multi-scale Reconstruction, we train the hierarchical architecture of 5-layer ResNet by each scale, each with 2000 steps. Network and parameters are adapted from the original SinGAN paper. We use the Adam optimizer with learning rate 0.0005, $\beta_1 = 0.5, \beta_2 = 0.999$, and we apply Cosine Annealing to update the learning rate. To stabilize the training, we used WGAN-GP[9] to regularize the loss with gradient penalty.

5 Conclusion

A CNN-based image manipulation model is proposed, which incorporates geometric constraints. In practice, user performs editing through geometric deformation on the contour representation of the image, and the model produces high-quality robust reconstructions. Since we perform target-specific post-processing technique that does not require supervision, the model shows improvement in terms of transferability over existing work. Although our approach captures objects' textures automatically even if they are not a priori seen by the neural network in the training database, still more complex real-world image data (e.g. the BSDS database) are not within the scope. Future work includes adapting to more diverse dataset in the real life.

References

1. Arbelaez, P., Maire, M., Fowlkes, C., Malik, J.: Contour detection and hierarchical image segmentation. IEEE Trans. Pattern Anal. Mach. Intell. **33**(5), 898–916 (May 2011). https://doi.org/10.1109/TPAMI.2010.161, http://dx.doi.org/10.1109/TPAMI.2010.161
2. Brendel, W., Bethge, M.: Approximating cnns with bag-of-local-features models works surprisingly well on imagenet. In: International Conference on Learning Representations (2018)
3. Chen, W., Hays, J.: Sketchygan: Towards diverse and realistic sketch to image synthesis. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 9416–9425 (2018)
4. Dekel, T., Gan, C., Krishnan, D., Liu, C., Freeman, W.T.: Sparse, smart contours to represent and edit images. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 3511–3520 (2018)
5. Dollár, P., Zitnick, C.L.: Fast edge detection using structured forests. IEEE transactions on pattern analysis and machine intelligence **37**(8), 1558–1570 (2014)
6. Dong, C., Loy, C.C., He, K., Tang, X.: Image super-resolution using deep convolutional networks. IEEE transactions on pattern analysis and machine intelligence **38**(2), 295–307 (2015)
7. Ghorbani, A., W.J.Z.J.K.B.: Towards automatic concept-based explanations. In: in: Wallach, H.M., Larochelle, H., Beygelzimer, A., dAlch-Buc, F., Fox, E.B., Garnett, R. (Eds.), Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada. p. 92739282 (2019)
8. Guidotti, R., Monreale, A., Matwin, S., Pedreschi, D.: Black box explanation by learning image exemplars in the latent feature space. In: Joint European Conference on Machine Learning and Knowledge Discovery in Databases. pp. 189–205. Springer (2019)
9. Gulrajani, I., Ahmed, F., Arjovsky, M., Dumoulin, V., Courville, A.C.: Improved training of wasserstein gans. In: Advances in neural information processing systems. pp. 5767–5777 (2017)
10. Ha, D., Eck, D.: A neural representation of sketch drawings. In: International Conference on Learning Representations (2018)

11. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 770–778 (2016)
12. Isola, P., Zhu, J.Y., Zhou, T., Efros, A.A.: Image-to-image translation with conditional adversarial networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 1125–1134 (2017)
13. Johnson, J., Alahi, A., Fei-Fei, L.: Perceptual losses for real-time style transfer and super-resolution. In: European conference on computer vision. pp. 694–711. Springer (2016)
14. Liu, R., Yu21, Q., Yu, S.X.: Unsupervised sketch to photo synthesis
15. Parekh, J., Mozharovskyi, P., d'Alche Buc, F.: A framework to learn with interpretation. arXiv preprint arXiv:2010.09345 (2020)
16. Parkhi, O.M., Vedaldi, A., Zisserman, A.: Deep face recognition. In: British Machine Vision Conference (2015)
17. Ronneberger, O., Fischer, P., Brox, T.: U-net: Convolutional networks for biomedical image segmentation. In: International Conference on Medical image computing and computer-assisted intervention. pp. 234–241. Springer (2015)
18. Sangkloy, P., Burnell, N., Ham, C., Hays, J.: The sketchy database: Learning to retrieve badly drawn bunnies. ACM Transactions on Graphics (proceedings of SIGGRAPH) (2016)
19. Sangkloy, P., Lu, J., Fang, C., Yu, F., Hays, J.: Scribbler: Controlling deep image synthesis with sketch and color. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 5400–5409 (2017)
20. Santurkar, S., Tsipras, D., Ilyas, A., Madry, A.: How does batch normalization help optimization? In: Proceedings of the 32nd International Conference on Neural Information Processing Systems. pp. 2488–2498 (2018)
21. Shaham, T.R., Dekel, T., Michaeli, T.: Singan: Learning a generative model from a single natural image. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 4570–4580 (2019)
22. Shocher, A., Bagon, S., Isola, P., Irani, M.: Ingan: Capturing and remapping the" dna" of a natural image. arXiv preprint arXiv:1812.00231 (2018)
23. Shocher, A., Cohen, N., Irani, M.: zero-shot super-resolution using deep internal learning. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 3118–3126 (2018)
24. Webster, R., Rabin, J., Simon, L., Jurie, F.: Detecting overfitting of deep generative networks via latent recovery. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 11273–11282 (2019)
25. Yang, F., Yang, H., Fu, J., Lu, H., Guo, B.: Learning texture transformer network for image super-resolution. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 5791–5800 (2020)
26. Yu, A., Grauman, K.: Fine-grained visual comparisons with local learning. In: Computer Vision and Pattern Recognition (CVPR) (Jun 2014)
27. Yu, Q., Liu, F., SonG, Y.Z., Xiang, T., Hospedales, T., Loy, C.C.: Sketch me that shoe. In: Computer Vision and Pattern Recognition (2016)
28. Zhu, J.Y., Krähenbühl, P., Shechtman, E., Efros, A.A.: Generative visual manipulation on the natural image manifold. In: European conference on computer vision. pp. 597–613. Springer (2016)
29. Zhu, J.Y., Park, T., Isola, P., Efros, A.A.: Unpaired image-to-image translation using cycle-consistent adversarial networks. In: Proceedings of the IEEE international conference on computer vision. pp. 2223–2232 (2017)