Geometric Deformation on Objects: Unsupervised Image Manipulation via Conjugation

Changqing Fu

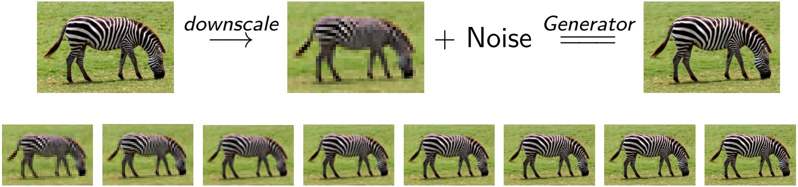
CEREMADE Winter School, Normandy

March 2, 2022

Practical Problem

- Problem Setting: “Image-to-Image Translation” (Pix2pix, 2017)
- Solution: Image Manipulation = Sparse Reconstruction + Multi-Scale Post-Processing
- Pros: Transferability and training data efficiency.

Practical Problem



- Problem Setting: "Image-to-Image Translation" (Pix2pix, 2017)
- Solution: Image Manipulation = Sparse Reconstruction + Multi-Scale Post-Processing
- Pros: Transferability and training data efficiency.

Practical Problem

- Problem Setting: "Image-to-Image Translation" (Pix2pix, 2017)
- Solution: Image Manipulation = Sparse Reconstruction + Multi-Scale Post-Processing
- Pros: Transferability and training data efficiency.

Math Formulation

$$\begin{tikzcd} \mathcal{M} \arrow[r, "A"] \arrow[d, "\varphi"'] & \mathcal{M} \\ \mathcal{N} \arrow[r, "B"] & \mathcal{N} \arrow[u, "\varphi^{-1}"'] \end{tikzcd}$$

- Conjugation $A = \varphi^{-1} \circ B \circ \varphi$, φ is a fixed Edge Detector
- Well-definedness: Is φ an *invertible* diffeomorphism?
- Sparsity of $\mathcal{N}$
- $\varphi^{-1} : \mathcal{N} \longrightarrow \mathcal{M}$, $\varphi^{-1}(y) := \{x \in \mathcal{M} | \varphi(x) = y\}, \forall y \in \mathcal{M}$ not unique!

Math Formulation

$$\begin{tikzcd} \mathcal{M} \arrow[r, "A"] \arrow[d, "\varphi"'] & \mathcal{M} \\ \mathcal{N} \arrow[r, "B"] & \mathcal{N} \arrow[u, "\varphi^{-1}"'] \end{tikzcd}$$

$$(\Omega, \mathcal{F}, \{\mathcal{D}_{ydata}, \mathcal{D}_G\})$$

$$(x, y) \sim \mathcal{D}_{data}, G(x)|x \sim \mathcal{D}_G, x \sim \pi(\cdot) = \mathcal{D}_{xdata}$$

- Conjugation $A = \varphi^{-1} \circ B \circ \varphi$, φ is a fixed Edge Detector
- Well-definedness: Is φ an *invertible* diffeomorphism?
- Sparsity of $\mathcal{N}$
- $\varphi^{-1} : \mathcal{N} \longrightarrow \mathcal{M}$, $\varphi^{-1}(y) := \{x \in \mathcal{M} | \varphi(x) = y\}, \forall y \in \mathcal{M}$ not unique!

Surrogate solution: GAN, a Data-driven approach.

$$G := \widehat{\varphi^{-1}} : \mathcal{N} \longrightarrow \mathcal{M}$$

$$s.t. \quad G = \text{argmin}_G - \int_\Omega \log \underbrace{L_D(G(x))}_{G(x) is\ real} \mathrm{d}\pi(x) + \lambda \mathbb{E}_{(x,y)\sim\mathcal{D}_{data}} \|G(x) - y\|_{L^1}$$

Blueprint

Lv	**Field**	**Particle**	**Motion**
I	Image Canvas Ω	Pixel (x_1, x_2)	Active Cont. $\gamma \in L^2([0,1];\Omega)$
II	Image Sp $L^2(\Omega, \mathbb{R}^3)$	Image u	$u(t,x) \in L^2([0,1]; L^2(\Omega, \mathbb{R}^3))$
II	Conv. Neural Net	Inner Features	$u_{t_i}(x) \in L^2(\Omega, \mathbb{R}^{d(t_i)})$
III	Parameter Space Θ	Linear filters θ	Deep Learning θ_t
IV	Dual Param Sp $\mathcal{W}$	GAN Discrim$w$	Deep Metric Learn $L_{D_{w_t}}(\cdot)$
V	Evaluation Measures	Metric $u \mapsto \mathcal{D}(u)$	e.g. Pre-training FID

Table: Hierarchy of Dynamics

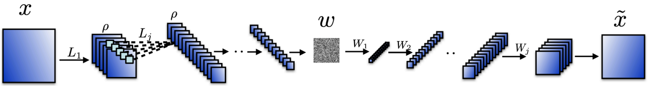


Figure: Homogeneous Filter on Patches

Dynamics of Points, Lines, Images

- Motion of points on images / Active Contor (Peyré, Cohen et al. 2011 MAL) $\gamma \in C([0,1];\mathbb{R}^2)$

$$\Phi(\gamma) = -\int_0^1 U(\gamma)\|\dot{\gamma}\|\mathrm{d}t + \lambda \int_0^1 \|\dot{\gamma}\| + \mu\|\ddot{\gamma}\|\mathrm{d}t$$

  Take $\mu = 0$, $W = U + \lambda$, γ is solved with the zero *level set* of φ's asymptotic limit

$$\frac{\mathrm{d}}{\mathrm{d}t}\varphi_t = \|\nabla\varphi_t\|\mathrm{div}\left(W\frac{\nabla\varphi_t}{\|\nabla\varphi_t\|}\right)$$

  momentum $T_\Phi = \frac{\gamma'(s)}{\|\gamma'(s)\|} \equiv 1$(Noether)

- Image morphing (Trouvé et al. 2005) $u \in L^2([0,1];L^2(\Omega,\mathbb{R}^3))$

$$\Phi(u) = \inf_{z=\dot{u}+\nabla u\cdot v} \int_0^1 \int_\Omega L^m[v,v] + \frac{1}{\delta}z^2\mathrm{d}x\mathrm{d}t$$

Dynamics of Network Layers

Sparse Reconstruction

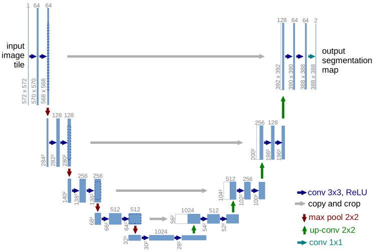


Figure: U-net (Olaf et al. 2015)

$$x_{n+1} = Encoder_n(x_n)$$
$$y_N = Encoder_n(x_N)$$
$$y_{n-1} = Decoder_n([x_n, y_n])$$

Back-propagation

Multi-Scale Reconstruction

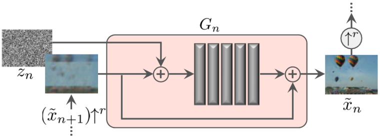


Figure: ResNet sturcture (Shaham et al. 2019)

$$x_{n-1} = x_n^{\uparrow} + Decoder_n(x_n + z_{n-1})$$

Learn-by-scale: Input/Output at arbitrary scale

$\boxed{\text{Neural Image Perception}}$

Dynamics of Scale Space

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

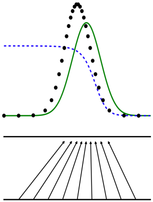
$$
\begin{aligned}
G(z) \quad &:= \quad \mathbf{G}_N(z_0, \ldots, z_N) \\
&= \quad \underbrace{\underbrace{G_0(z_0)}_{\triangleq x_0} + G_1(G_0(z_0) + z_1)}_{\triangleq x_1} + \cdots \\
&\qquad + G_N(\underbrace{G_{N-1}(\underbrace{\cdots G_1(G_0(z_0) + z_1) \cdots}_{\triangleq x_{N-2}} + z_{N-1})}_{\triangleq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

Dynamics of GAN Parameters I

Fix discriminator D: a (learned) metric of the image being "fake".

$$\Phi_D(G) = \mathbb{E}_{y\sim\mathcal{D}_{ydata}}[\log D(y)] + \mathbb{E}_{x\sim\pi}[\log(1 - D(G(x)))]$$

- Dynamics (Training / Optimization / Fixed-point Method)
- $y \sim \mathcal{D}_{ydata}, x \sim \mathcal{N}(0,1), G \equiv f(y|x)$ ("Sampling via Inverse transform F^{-1}" / Importance Sampling / Change of variable / Bayesian Prior / Radon-Nikodym...)



$$D^*(y) = \frac{p_{ydata}(y)}{p_{ydata}(y) + p_G(y)}$$

(Picture from Goodfellow 2014)

Dynamics of GAN Parameters I

Fix discriminator D: a (learned) metric of the image being “fake”.

$$\Phi_D(G) = \mathbb{E}_{y\sim\mathcal{D}_{ydata}}[\cancel{\log} D^2(y)] + \mathbb{E}_{x\sim\pi}[\cancel{\log}(1 - D(G(x)))^2]$$

- Dynamics (Training / Optimization / Fixed-point Method)
- $y \sim \mathcal{D}_{ydata}, x \sim \mathcal{N}(0,1)$, $G \equiv f(y|x)$ (“Sampling via Inverse transform F^{-1}” / Importance Sampling / Change of variable / Bayesian Prior / Radon-Nikodym...)

Variants:

- If $x \sim \mathcal{D}_{xdata}$, G is called a “Conditional GAN”.
- $G = Decoder \circ Encoder$, $z = Encoder(x)$,$y = Decoder(z)$, $z|x \sim \mathcal{N}(0,1)$ “Variational Auto-encoder (VAE)”.

Dynamics of GAN Parameters II

Min-max game equilibrium / Saddle-point / Alternating Direction

$$\min_G \max_D \Phi_D(G) = \mathbb{E}_y[D(y)] - \mathbb{E}_x[D(G(x))]$$

$$F_h(G, D) = (G, D) - hv(G, D), \text{where } v(G, D) = \begin{pmatrix} \nabla_G \Phi \\ -\nabla_D \Phi \end{pmatrix}$$

$$\begin{pmatrix} \dot{G} \\ \dot{D} \end{pmatrix} = \begin{pmatrix} -\nabla_G \Phi \\ \nabla_D \Phi \end{pmatrix}$$

Structure: $\Pi_{layers}(\text{Activation}_{\text{element-wise}} \circ \text{Linear}_{\text{centered}})$, w_i are wavelet filters

$$NN(x) = \rho_L(w_L \cdots \rho_1(w_1 x))$$

Gradient Penalty: 1-Lipshitz

$$\lambda \mathbb{E}_{x \sim \mu_\theta}(\|\nabla_x f_w(x)\|_2 - 1)^2$$

Evaluation of GAN

	Measure	Description
Quantitative	1. Average Log-likelihood [18, 22]	• Log likelihood of explaining realworld held out/test data using a density estimated from the generated data (*e.g.* using KDE or Parzen window estimation). $L = \frac{1}{N}\sum_i \log P_{model}(\mathbf{x}_i)$
	2. Coverage Metric [33]	• The probability mass of the true data "covered" by the model distribution $C := P_{data}(dP_{model} > t)$ with t such that $P_{model}(dP_{model} > t) = 0.95$
	3. Inception Score (IS) [3]	• KLD between conditional and marginal label distributions over generated data. $\exp\left(\mathbb{E}_{\mathbf{x}}\left[\mathbb{KL}\left(p\left(y \mid \mathbf{x}\right) \Vert p\left(y\right)\right)\right]\right)$
	4. Modified Inception Score (m-IS) [34]	• Encourages diversity within images sampled from a particular category. $\exp(\mathbb{E}_{\mathbf{x}_i}[\mathbb{E}_{\mathbf{x}_j}[(\mathbb{KL}(P(y\mid\mathbf{x}_i)\Vert P(y\mid\mathbf{x}_j))]])$
	5. Mode Score (MS) [35]	• Similar to IS but also takes into account the prior distribution of the labels over real data. $\exp\left(\mathbb{E}_{\mathbf{x}}\left[\mathbb{KL}\left(p\left(y \mid \mathbf{x}\right) \Vert p\left(y^{train}\right)\right)\right] - \mathbb{KL}\left(p\left(y\right) \Vert p\left(y^{train}\right)\right)\right)$
	6. AM Score [36]	• Takes into account the KLD between distributions of training labels vs. predicted labels, as well as the entropy of predictions. $\mathbb{KL}(p(y^{\mathrm{train}}) \Vert p(y)) + \mathbb{E}_{\mathbf{x}}[H(y\mid\mathbf{x})]$
	7. Fréchet Inception Distance (FID) [37]	• Wasserstein-2 distance between multi-variate Gaussians fitted to data embedded into a feature space $\boxed{FID(r,g) = \Vert\mu_r - \mu_g\Vert_2^2 + Tr(\Sigma_r + \Sigma_g - 2(\Sigma_r\Sigma_g)^{\frac{1}{2}})}$
	8. Maximum Mean Discrepancy (MMD) [38]	• Measures the dissimilarity between two probability distributions P_r and P_g using samples drawn independently from each distribution. $M_k(P_r, P_g) = \mathbb{E}_{\mathbf{x},\mathbf{x}'\sim P_r}[k(\mathbf{x},\mathbf{x}')] - 2\mathbb{E}_{\mathbf{x}\sim P_r,\boldsymbol{y}\sim P_g}[k(\mathbf{x},\boldsymbol{y})] + \mathbb{E}_{\boldsymbol{y},\boldsymbol{y}'\sim P_g}[k(\boldsymbol{y},\boldsymbol{y}')]$
	9. The Wasserstein Critic [39]	• The critic (*e.g.* an NN) is trained to produce high values at real samples and low values at generated samples $\hat{W}(\mathbf{x}_{test}, \mathbf{x}_g) = \frac{1}{N}\sum_{i=1}^{N}\hat{f}(\mathbf{x}_{test}[i]) - \frac{1}{N}\sum_{i=1}^{N}\hat{f}(\mathbf{x}_g[i])$
	10. Birthday Paradox Test [27]	• Measures the support size of a discrete (continuous) distribution by counting the duplicates (near duplicates)
	11. Classifier Two Sample Test (C2ST) [40]	• Answers whether two samples are drawn from the same distribution (*e.g.* by training a binary classifier)
	12. Classification Performance [1, 15]	• An indirect technique for evaluating the quality of unsupervised representations (*e.g.* feature extraction; FCN score). See also the GAN Quality Index (GQI) [41].
	13. Boundary Distortion [42]	• Measures diversity of generated samples and covariate shift using classification methods.
	14. Number of Statistically-Different Bins (NDB) [43]	• Given two sets of samples from the same distribution, the number of samples that fall into a given bin should be the same up to sampling noise
	15. Image Retrieval Performance [44]	• Measures the distributions of distances to the nearest neighbors of some query images (*i.e.* diversity)
	16. Generative Adversarial Metric (GAM) [31]	• Compares two GANs by having them engaged in a battle against each other by swapping discriminators or generators. $p(\mathbf{x}\mid y=1; M_1^{'})/p(\mathbf{x}\mid y=1; M_2^{'}) = \left(p(y=1\mid\mathbf{x}; D_1)p(\mathbf{x}; G_2)\right)/\left(p(y=1\mid\mathbf{x}; D_2)p(\mathbf{x}; G_1)\right)$
	17. Tournament Win Rate and Skill Rating [45]	• Implements a tournament in which a player is either a discriminator that attempts to distinguish between real and fake data or a generator that attempts to fool the discriminators into accepting fake data as real.
	18. Normalized Relative Discriminative Score (NRDS) [32]	• Compares n GANs based on the idea that if the generated samples are closer to real ones, more epochs would be needed to distinguish them from real samples.
	19. Adversarial Accuracy and Divergence [46]	• Adversarial Accuracy. Computes the classification accuracies achieved by the two classifiers, one trained on real data and another on generated data, on a labeled validation set to approximate $P_g(y\mid\mathbf{x})$ and $P_r(y\mid\mathbf{x})$. Adversarial Divergence: Computes $\mathbb{KL}(P_g(y\mid\mathbf{x}), P_r(y\mid\mathbf{x}))$
	20. Geometry Score [47]	• Compares geometrical properties of the underlying data manifold between real and generated data.
	21. Reconstruction Error [48]	• Measures the reconstruction error (*e.g.* L_2 norm) between a test image and its closest generated image by optimizing for z (*i.e.* $min_{\mathbf{z}}\Vert G(\mathbf{z}) - \mathbf{x}^{(test)}\Vert^2$)
	22. Image Quality Measures [49, 50, 51]	• Evaluates the quality of generated images using measures such as SSIM, PSNR, and sharpness difference
	23. Low-level Image Statistics [52, 53]	• Evaluates how similar low-level statistics of generated images are to those of natural scenes in terms of mean power spectrum, distribution of random filter responses, contrast distribution, etc.
	24. Precision, Recall and F_1 score [23]	• These measures are used to quantify the degree of overfitting in GANs, often over toy datasets.
Qualitative	1. Nearest Neighbors	• To detect overfitting, generated samples are shown next to their nearest neighbors in the training set
	2. Rapid Scene Categorization [18]	• In these experiments, participants are asked to distinguish generated samples from real images in a short presentation time (*e.g.* 100 ms); *i.e.* real v.s fake
	3. Preference Judgment [54, 55, 56, 57]	• Participants are asked to rank models in terms of the fidelity of their generated images (*e.g.* pairs, triples)
	4. Mode Drop and Collapse [58, 59]	• Over datasets with known modes (*e.g.* a GMM or a labeled dataset), modes are computed as by measuring the distances of generated data to mode centers
	5. Network Internals [1, 60, 61, 62, 63, 64]	• Regards exploring and illustrating the internal representation and dynamics of models (*e.g.* space continuity) as well as visualizing learned features

Figure: $IS = \exp(\mathbb{E}_{x\sim\mu_\theta} KL(p(y|x)\|p(y)))$

$$FID = \|\mu - \mu_r\|_2^2 + \mathrm{tr}(\Sigma + \Sigma_r - 2(\Sigma^{1/2}\Sigma_r\Sigma^{1/2})^{1/2})$$

$$MMD = \left\|\frac{1}{n}\sum_{i=1}^{n}\varphi(x_i) - \frac{1}{m}\sum_{i=1}^{m}\varphi(y_i)\right\|_{\mathcal{H}}^2 = \frac{1}{n^2}\sum_{i=1}^{n}\sum_{j=1}^{n}k(x_i, x_j) + \frac{1}{m^2}\sum_{i=1}^{m}\sum_{j=1}^{m}k(y_i, y_j) - \frac{2}{nm}\sum_{i=1}^{n}\sum_{j=1}^{m}k(x_i, y_j)$$

Properties of CNN

- Center preserving $\int_\Omega u \mathrm{d}x \equiv \mathit{const.}$: StyleGAN (Center$\sim$Image Force)
- Norm preserving $\int_\Omega u^2 \mathrm{d}x \equiv \mathit{const.}$: Batch / Instance / Layer / Group Normalization (Covariance$\sim$Style)
- Invertibility: Normalizing Flow
- Equivariance, conserved quantity, frequency separation $g \circ \rho = \rho \circ g$: StyleGAN3 (Alias-free)
- Non-homogeneity of local filters: Transformers
- Integrability across scales: This work (Space dimension is time dependent)

Applications

- Medical Image Synthesis
- Active Learning
- Privacy-Protective Learning
- Cross-Domain/Transfer Learning

Thank You

Q&A

*Funded in part by the French government under management of Agence Nationale de la Recherche as part of the "Investissements d'avenir" program, reference ANR-19-P3IA-0001 (PRAIRIE 3IA Institute).