Exact Feature Distribution Matching for Arbitrary Style Transfer and Domain Generalization

Yabin Zhang[1], Minghan Li[1], Ruihuang Li[1], Kui Jia[2], Lei Zhang[1*]
[1]Hong Kong Polytechnic University [2]South China University of Technology
{csybzhang,csrhli,cslzhang}@comp.polyu.edu.hk, liminghan0330@gmail.com, kuijia@scut.edu.cn

CVPR 2022

Reporter: 王启迅

2022/11/17

Main idea

Given two feature vectors $\boldsymbol{x}$ (content) and $\boldsymbol{y}$ (style), we want to transform $\boldsymbol{x}$ to $\boldsymbol{o}$, so that the statistics of $\boldsymbol{o}$ is equivalent to $\boldsymbol{y}$.

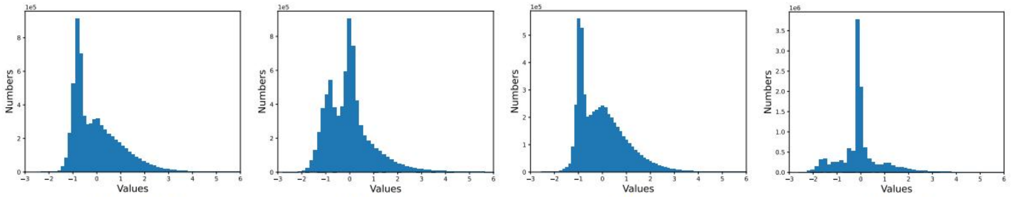
Statistics: mean, standard deviation, skewness (偏度), kurtosis (峰度)…

Related methods

1. AdaIN (**Adaptive instance normalization**): transform $x \in R^n$ (content) to an output vector $o \in R^n$ whose mean and standard deviation match those of $y \in R^m$ (style).

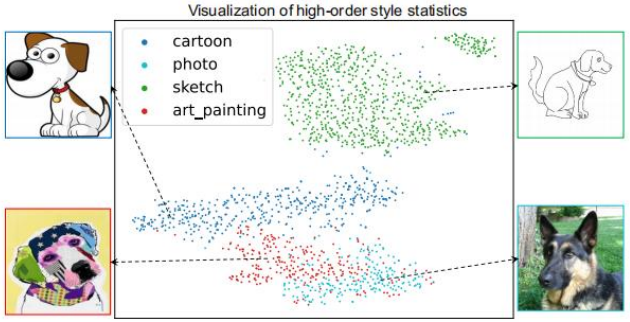
$$\boldsymbol{o} = \frac{\boldsymbol{x} - \mu(\boldsymbol{x})}{\sigma(\boldsymbol{x})}\sigma(\boldsymbol{y}) + \mu(\boldsymbol{y})$$

Problem: real-world data are usually too complicated to be modeled by Gaussian.



(a) Art painting (b) Cartoon (c) Photo (d) Sketch

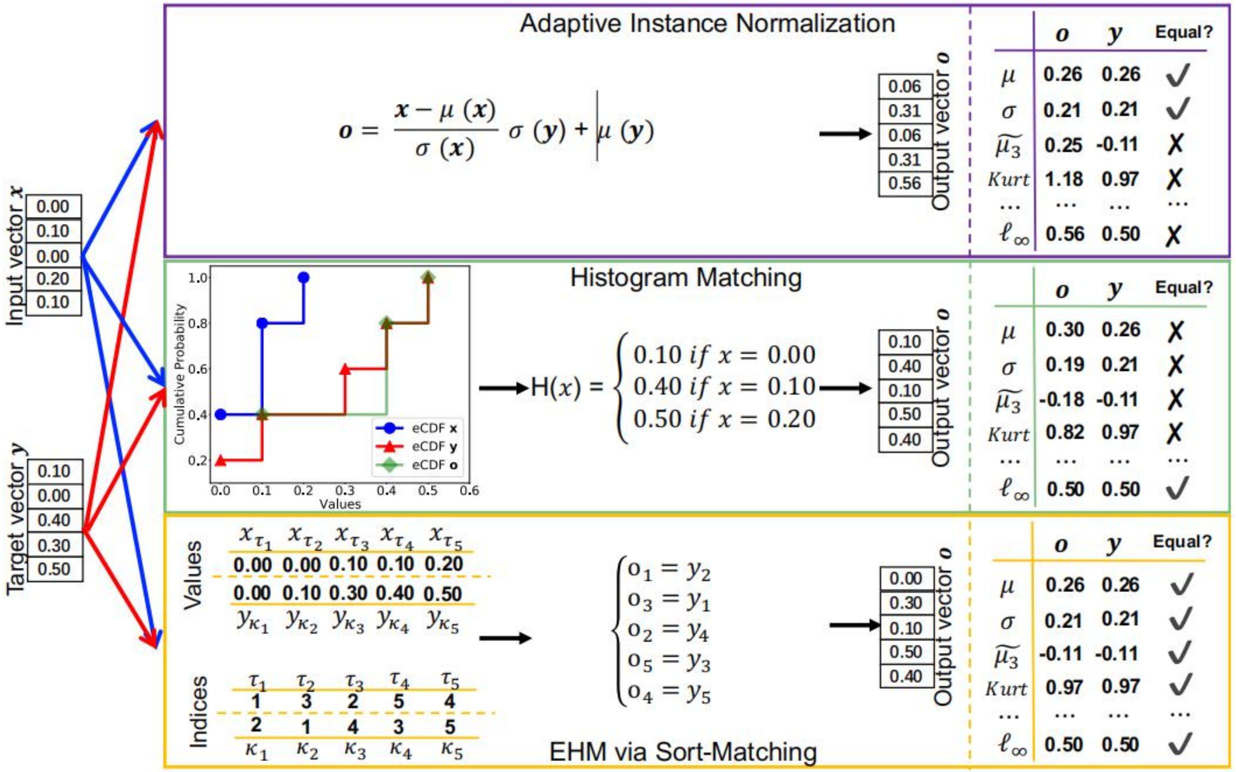
Features from the first block of RN-18. Normalize the mean and std to be 0 and 1 of each channel. Each value point is a feature value. All feature value of all samples are collected.



t-SNE [53] visualization of the third standardized moment-skewness. Style information can be represented by high-order statistics beyond mean and standard deviation.

Related methods

1. AdaIN $$\boldsymbol{o} = \frac{\boldsymbol{x} - \mu(\boldsymbol{x})}{\sigma(\boldsymbol{x})}\sigma(\boldsymbol{y}) + \mu(\boldsymbol{y})$$



Related methods

2. HM (**Histogram matching**): transform $x \in R^n$ (content) to an output vector $o \in R^n$ whose eCDF (empirical Cumulative Distribution Function) matches that of $y \in R^m$ (style).

eCDF of x and y: (x_i and y_i is the i-th element of the vector)

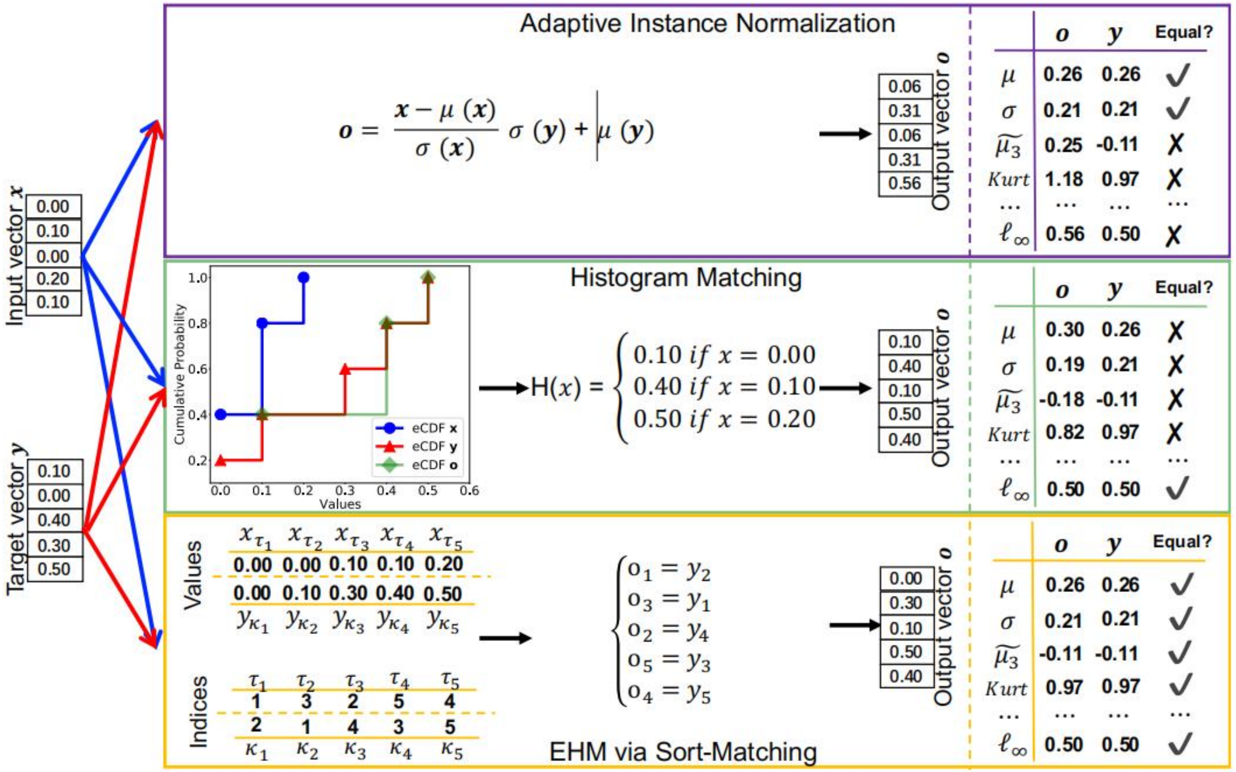
$$\widehat{F}_X(x) = \frac{1}{n}\sum_{i=1}^{n} \mathbf{1}_{x_i \le x}, \quad \widehat{F}_Y(y) = \frac{1}{m}\sum_{i=1}^{m} \mathbf{1}_{y_i \le y}$$

For each element x_i, find y_i that satisfies $\widehat{F}_X(x_i) = \widehat{F}_Y(y_j)$

Resulting the transformation: $H(x_i) = y_j$

Related methods

2. HM: $\widehat{F}_X(x_i) = \widehat{F}_Y(y_j) \;\rightarrow\; H(x_i) = y_j$



Problem: repeated feature values (0.00) of x will be compressed into a single point, leading to inaccurate eCDF.

Related methods

3. EHM (**Exact Histogram matching**): transform x into o, whose eCDF exactly matches y.

Sort x and y and get indexes (ascending order) $\boldsymbol{x} : \tau = \begin{pmatrix} \tau_1 & \tau_2 & \tau_3 & \cdots & \tau_n \end{pmatrix}$ $\boldsymbol{y} : \kappa = \begin{pmatrix} \kappa_1 & \kappa_2 & \kappa_3 & \cdots & \kappa_n \end{pmatrix}$

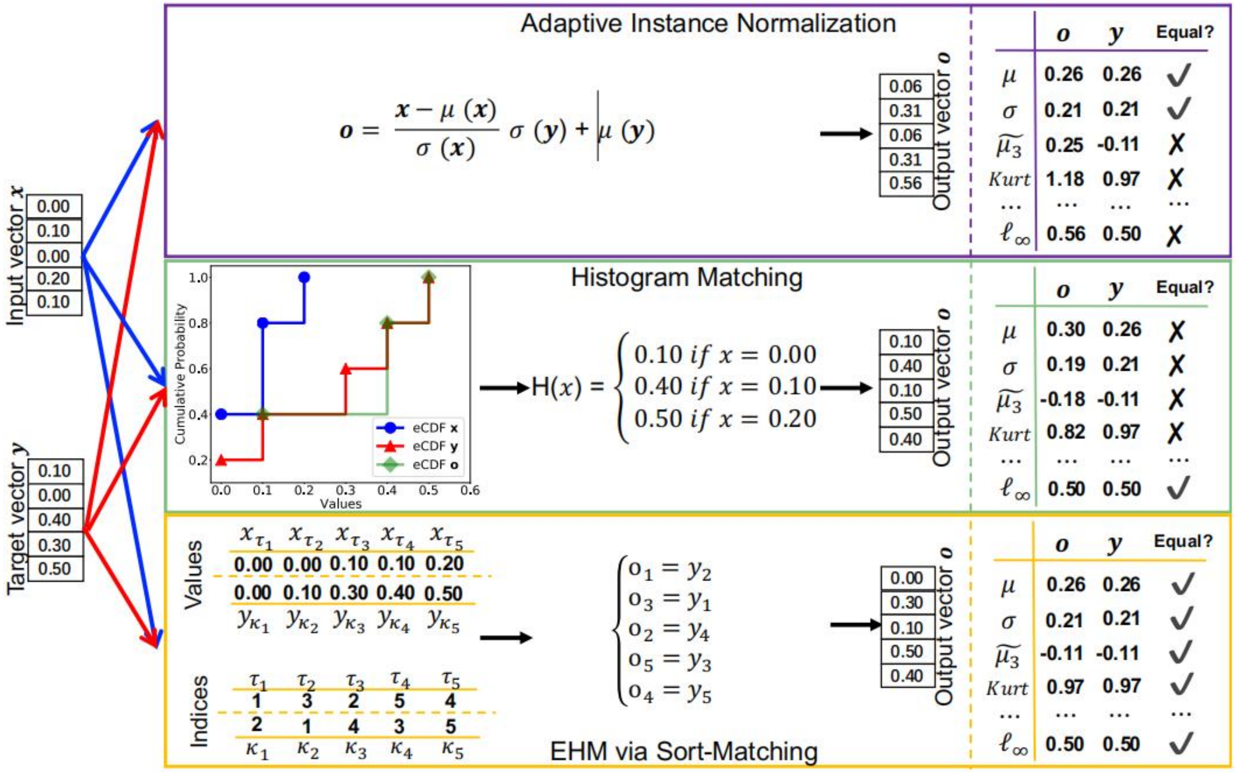
Where $\boldsymbol{x}_{\tau_i} \leq \boldsymbol{x}_{\tau_j}$ if $i < j$. Same for y.

Transformation: $o_{\tau_i} = \boldsymbol{y}_{\kappa_i}$

Comment: Content (the relative size relationship of the pixels) is maintained (x’ s pixel layout), style is replaced (y’ s pixel value)

Related methods

3. EHM: $\widehat{F}_X(x_i) = \widehat{F}_Y(y_j)$ $\;-H(x_i) = y_j$



Adapt EHM to style transfer and domain generalization

EHM: $o_{\tau_i} = y_{\kappa_i}$

To enable back-propagation: $\mathbf{EFDM}(\boldsymbol{x}, \boldsymbol{y}) : o_{\tau_i} = x_{\tau_i} + y_{\kappa_i} - \langle x_{\tau_i} \rangle$

<> Is the stop-gradient operation.

Algorithm 1 PyTroch-like pseudo-code for EFDM.

```
# x, y: input and target vectors of the same shape (n)
_, IndexX = torch.sort(x)                  # Sort x values
SortedY, _ = torch.sort(y)                 # Sort y values
InverseIndex = IndexX.argsort(-1)
return x + SortedY.gather(-1, InverseIndex) - x.detach()
```

Arbitrary style transfer

Fix an encoder f (first few layer of pre-trained VGG-19)

Content image X and style image Y [B,C,HW]

Stylized feature: $\boldsymbol{S} = \mathrm{EFDM}(f(\boldsymbol{X}), f(\boldsymbol{Y}))$

Train a randomly initialized decoder g

$g(S)$ is the reconstructed image with content of X and style of Y

Loss: $\mathcal{L} = \mathcal{L}_c + \omega \mathcal{L}_s$

Arbitrary style transfer

Content loss: $\mathcal{L}_c = \|f(g(\boldsymbol{S})) - \boldsymbol{S}\|_2$

Style loss: $\mathcal{L}_s = \sum_{i=1}^{L} \|\phi_i(g(\boldsymbol{S})) - \mathrm{EFDM}(\phi_i(g(\boldsymbol{S})), \phi_i(\boldsymbol{Y}))\|_2$

Match the high-order statistics of the reconstructed image with that of Y. (Why not using $\phi_i(Y)$ as the second)

ϕ_i is the layer i of VGG-19

Ps: Content/style loss in AdaIN: $\mathcal{L}_c = \|f(g(t)) - t\|_2$ $\quad$ $\mathcal{L}_s = \sum_{i=1}^{L} \|\mu(\phi_i(g(t))) - \mu(\phi_i(s))\|_2 + \sum_{i=1}^{L} \|\sigma(\phi_i(g(t))) - \sigma(\phi_i(s))\|_2$

Domain Generalization

Use EFGM to generate diverse features

$$\mathrm{EFDMix}(\boldsymbol{x}, \boldsymbol{y}) : o_{\tau_i} = x_{\tau_i} + (1-\lambda) y_{\kappa_i} - (1-\lambda)\langle x_{\tau_i} \rangle \qquad \lambda \sim \mathrm{Beta}(\alpha, \alpha)$$

The choice of style image Y:

- If domain label is available: Y is from other domains
- If domain label is not available: Y is randomly chosen

With probability of 0.5 to activate EFDMix in the forward pass

When $\lambda = 0$, use style image completely

Experiments on AST

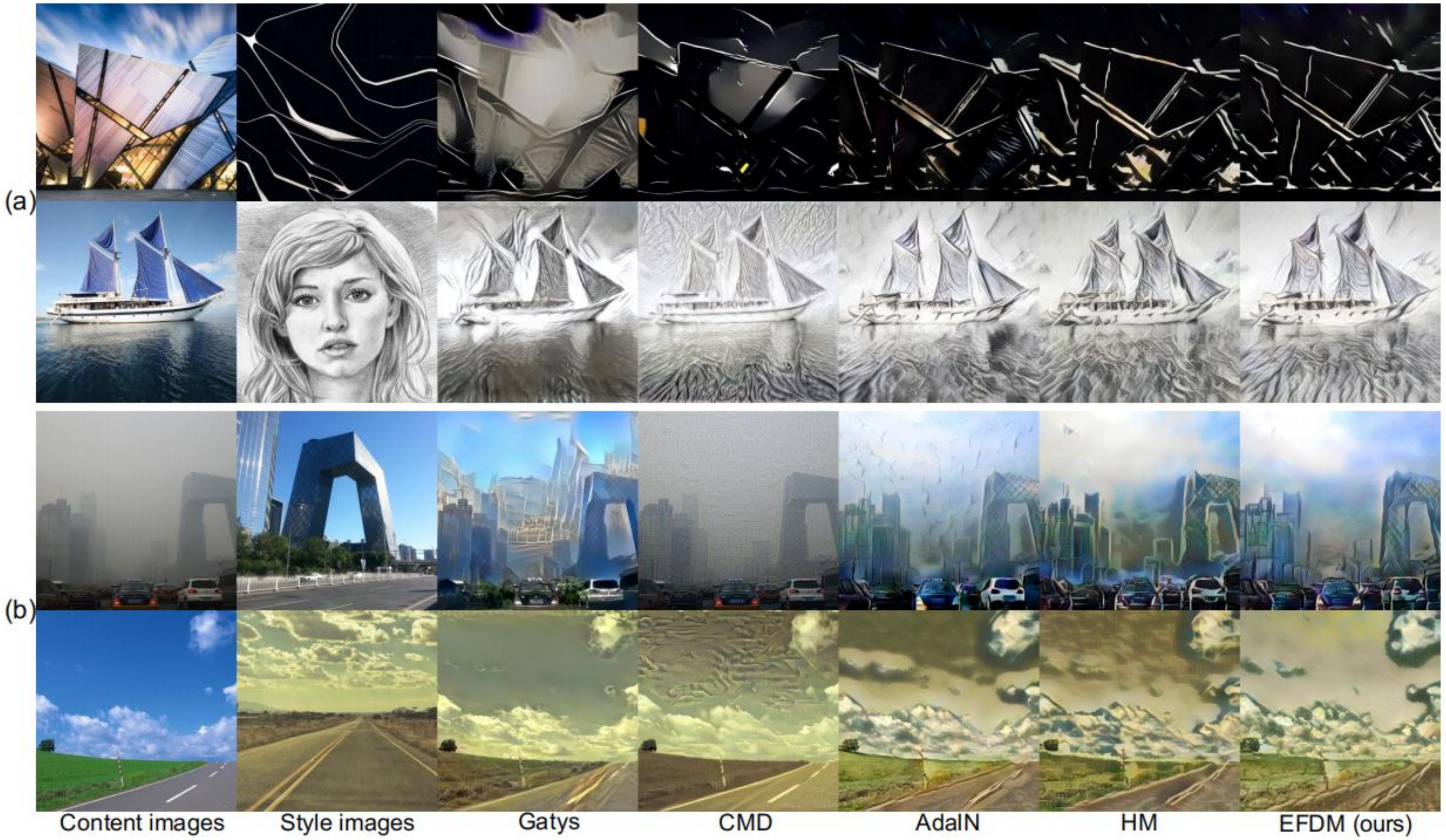


Figure 5. Illustration of results on (a) style transfer [21] (top two rows) and (b) the more challenging photo-realistic style transfer [38] (bottom two rows) tasks. Results of 'Gatys' [12] and 'CMD' [25] are obtained with official codes. For HM, we use HM, instead of EHM, to approximately match eCDFs. More visualizations are provided in the **supplementary file**.

Experiments on DG

Method	Art	Cartoon	Photo	Sketch	Avg
Leave-one-domain-out generalization results					
JiGen [3]	79.4	75.3	96.0	71.6	80.5
L2A-OT [71]	83.3	78.2	96.2	73.6	82.8
ResNet-18 [20]	77.0±0.6	75.9±0.6	96.0±0.1	69.2±0.6	79.5
+ Mixup [64]	76.8±0.7	74.9±0.7	95.8±0.3	66.6±0.7	78.5
+ MixStyle w/ domain label [72]	83.1±0.8	78.6±0.9	95.9±0.4	74.2±2.7	82.9
+ EFDMix w/ domain label (ours)	**83.9**±0.4	**79.4**±0.7	**96.8**±0.4	**75.0**±0.7	**83.9**
ResNet-50 [20]	84.4±0.9	77.1±1.4	97.6±0.2	70.8±0.7	82.5
+ MixStyle w/ domain label [72]	90.3±0.3	82.3±0.7	97.7±0.4	74.7±0.7	86.2
+ EFDMix w/ domain label (ours)	**90.6**±0.3	**82.5**±0.7	**98.1**±0.2	**76.4**±1.2	**86.9**
Single source generalization results					
ResNet-18 [20]	58.6±2.4	66.4±0.7	34.0±1.8	27.5±4.3	46.6
+ MixStyle w/ random shuffle [72]	61.9±2.2	71.5±0.8	41.2±1.8	32.2±4.1	51.7
+ EFDMix w/ random shuffle (ours)	**63.2**±2.3	**73.9**±0.7	**42.5**±1.8	**38.1**±3.7	**54.4**
ResNet-50 [20]	63.5±1.3	69.2±1.6	38.0±0.9	31.4±1.5	50.5
+ MixStyle w/ random shuffle [72]	73.2±1.1	74.8±1.1	46.0±2.0	40.6±2.0	58.6
+ EFDMix w/ random shuffle (ours)	**75.3**±0.9	**77.4**±0.8	**48.0**±0.9	**44.2**±2.4	**61.2**

Analysis on the role the different orders of the statistics

Figure 8. Qualitative analyses on the role of different orders of feature statistics on AST.

EFDM preserves most content information

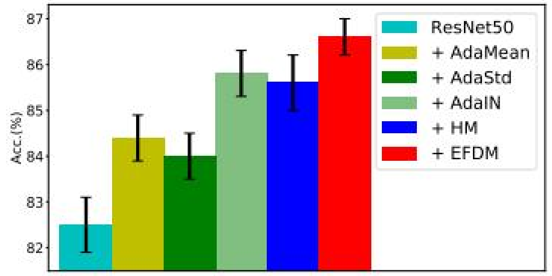


Figure 9. Quantitative analyses on the role of different orders of feature statistics on PACS dataset.