Kepco - UET

Trường Đại học Công nghệ

FINAL PROJECT

Presented by: Nguyễn Quý Đang - 22022500

OVERVIEW

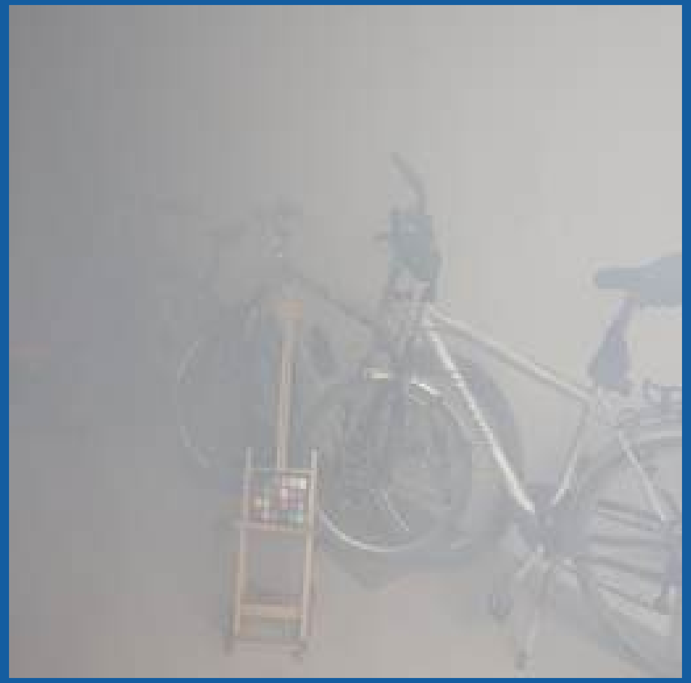
01 Dehazing

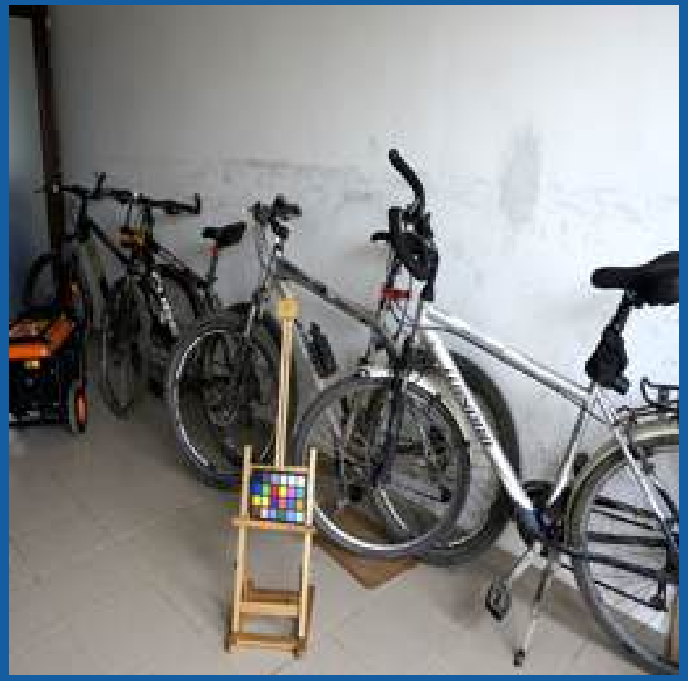
02 CycleGan

03 Virtual try on

04 VITION

01 Dehazing





02 CycleGAN

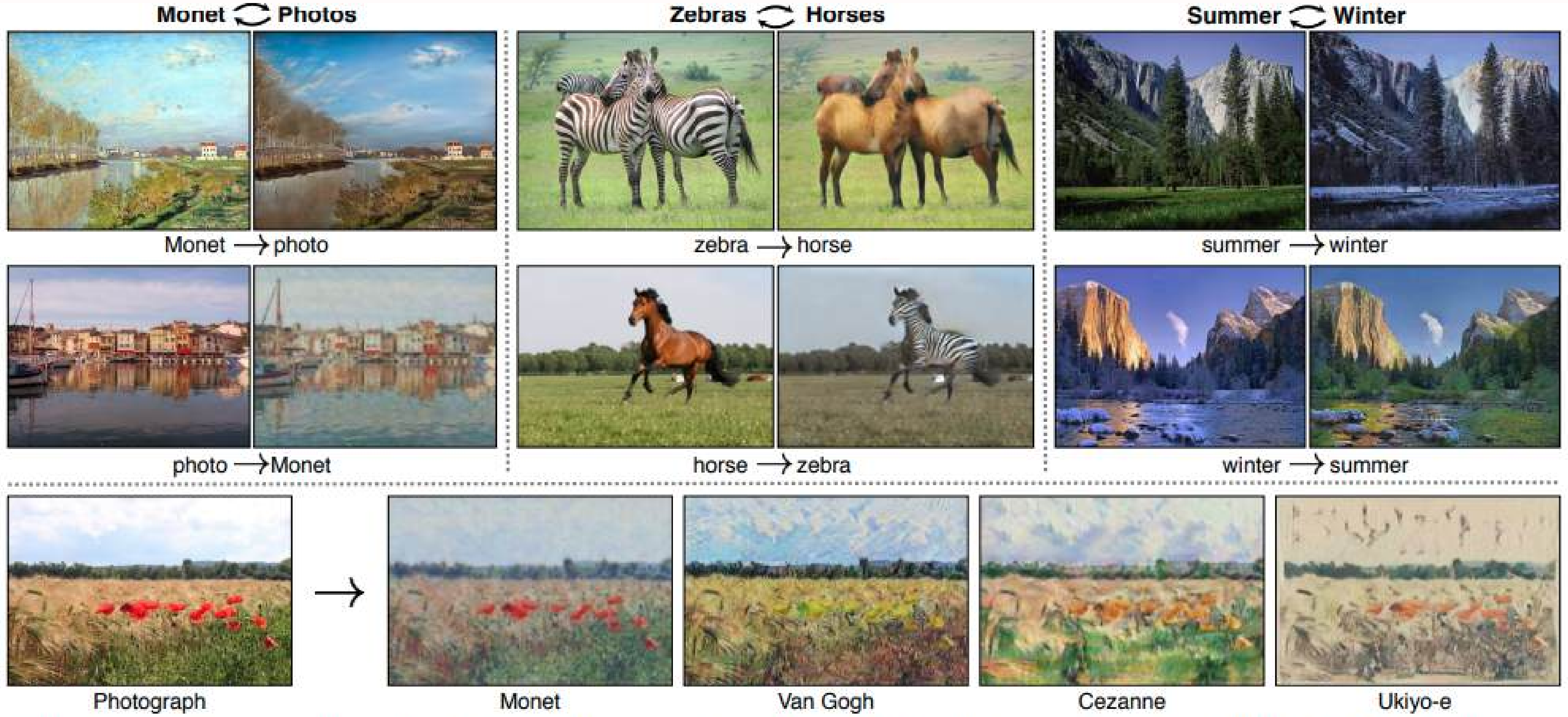


Figure 1: Given any two unordered image collections X and Y, our algorithm learns to automatically "translate" an image from one into the other and vice versa: *(left)* Monet paintings and landscape photos from Flickr; *(center)* zebras and horses from ImageNet; *(right)* summer and winter Yosemite photos from Flickr. Example application *(bottom)*: using a collection of paintings of famous artists, our method learns to render natural photographs into the respective styles.

02 CycleGAN

2.1 Architecture

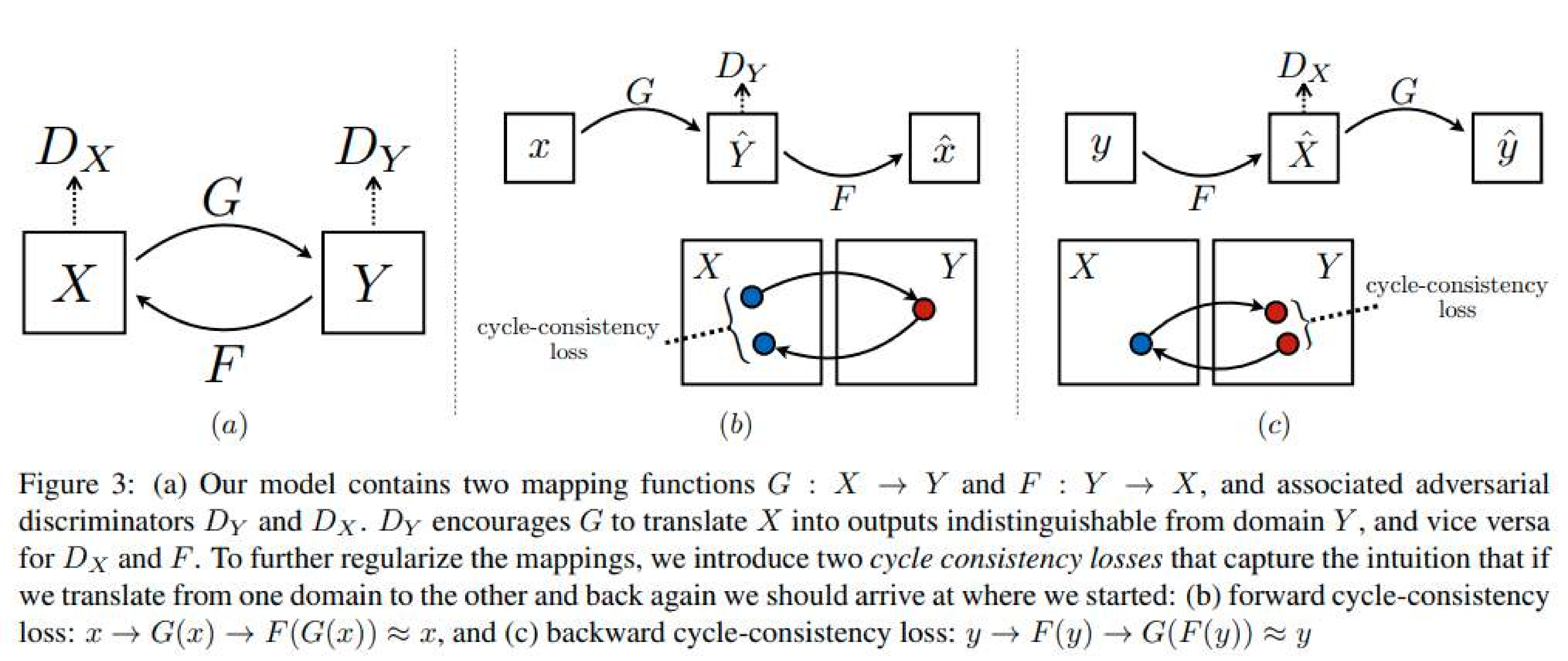
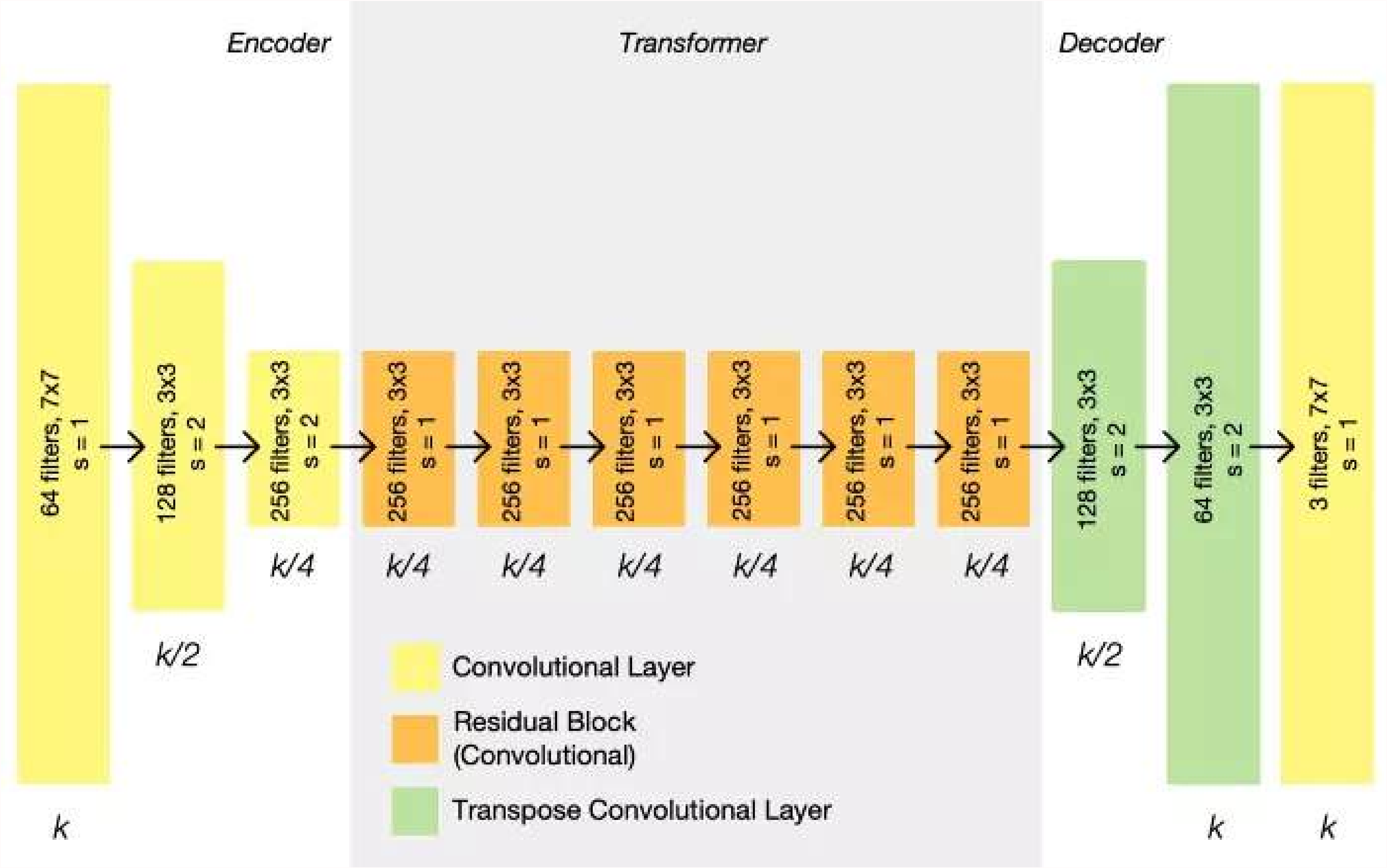


Figure 3: (a) Our model contains two mapping functions $G : X \to Y$ and $F : Y \to X$, and associated adversarial discriminators D_Y and D_X. D_Y encourages G to translate X into outputs indistinguishable from domain Y, and vice versa for D_X and F. To further regularize the mappings, we introduce two *cycle consistency losses* that capture the intuition that if we translate from one domain to the other and back again we should arrive at where we started: (b) forward cycle-consistency loss: $x \to G(x) \to F(G(x)) \approx x$, and (c) backward cycle-consistency loss: $y \to F(y) \to G(F(y)) \approx y$

02

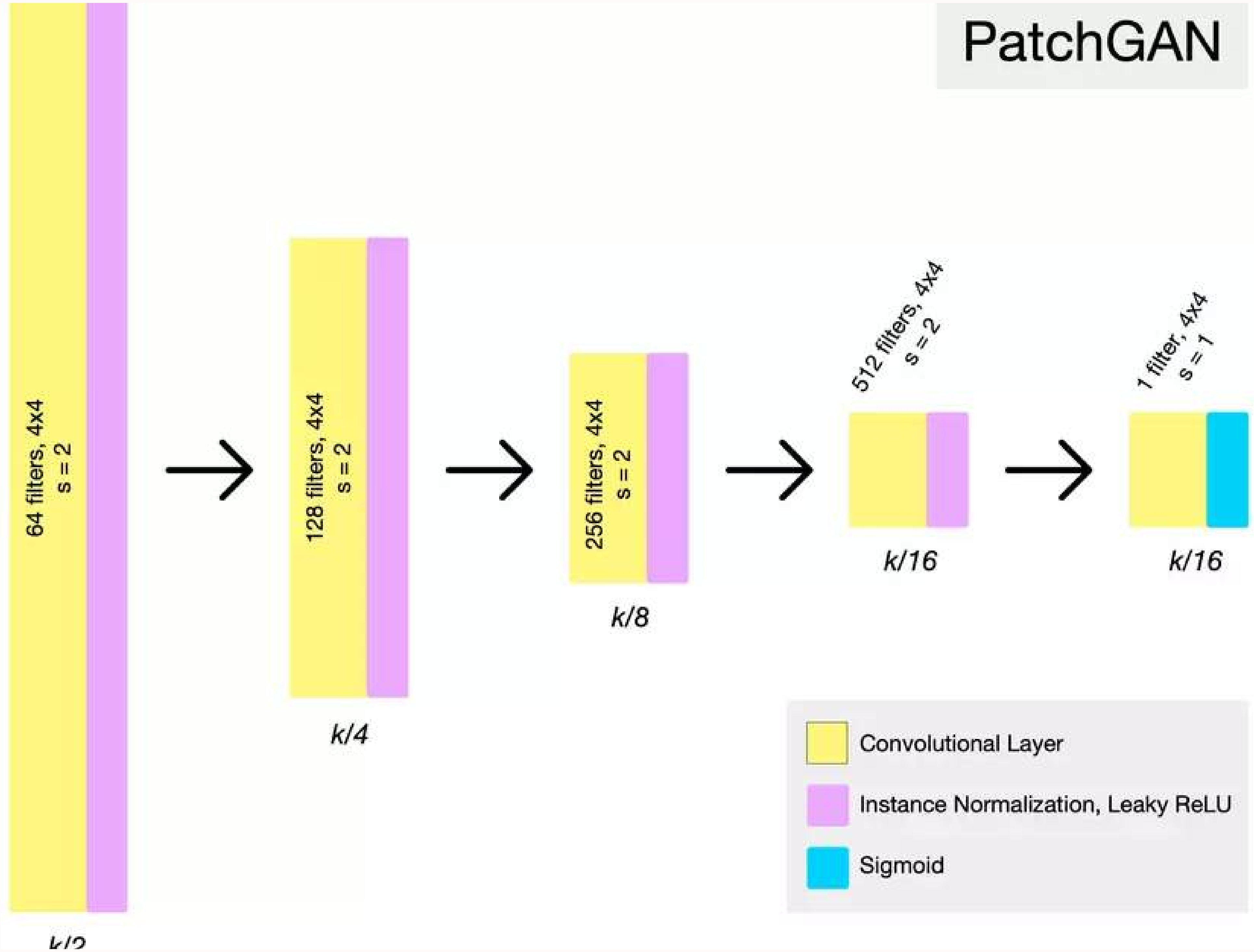
CycleGAN

2.2 Generator



02 CycleGAN

2.3 Discriminator



02 CycleGAN

2.4 Object function

$$L_{adv}(G, D_Y, X, Y) = \tfrac{1}{n}[log D_Y(y)] + \tfrac{1}{n}[log(1 - D_Y(G(x))]$$

$$L_{adv}(F, D_X, Y, X) = \tfrac{1}{n}[log D_X(x)] + \tfrac{1}{n}[log(1 - D_X(F(y))]$$

Adversarial Loss

$$x \rightarrow G(x) \rightarrow F(G(x)) \approx x$$

$$L_{cycle}(G, F) = \tfrac{1}{n}\sum |F(G(x_i)) - x_i| + |G(F(y_i)) - y_i|$$

Cycle Consistency Loss

$$L = L_{adv}(G, D_Y, X, Y) + L_{adv}(F, D_X, Y, X) + \lambda L_{cycle}(G, F)$$

02 CycleGAN

2.5 Result

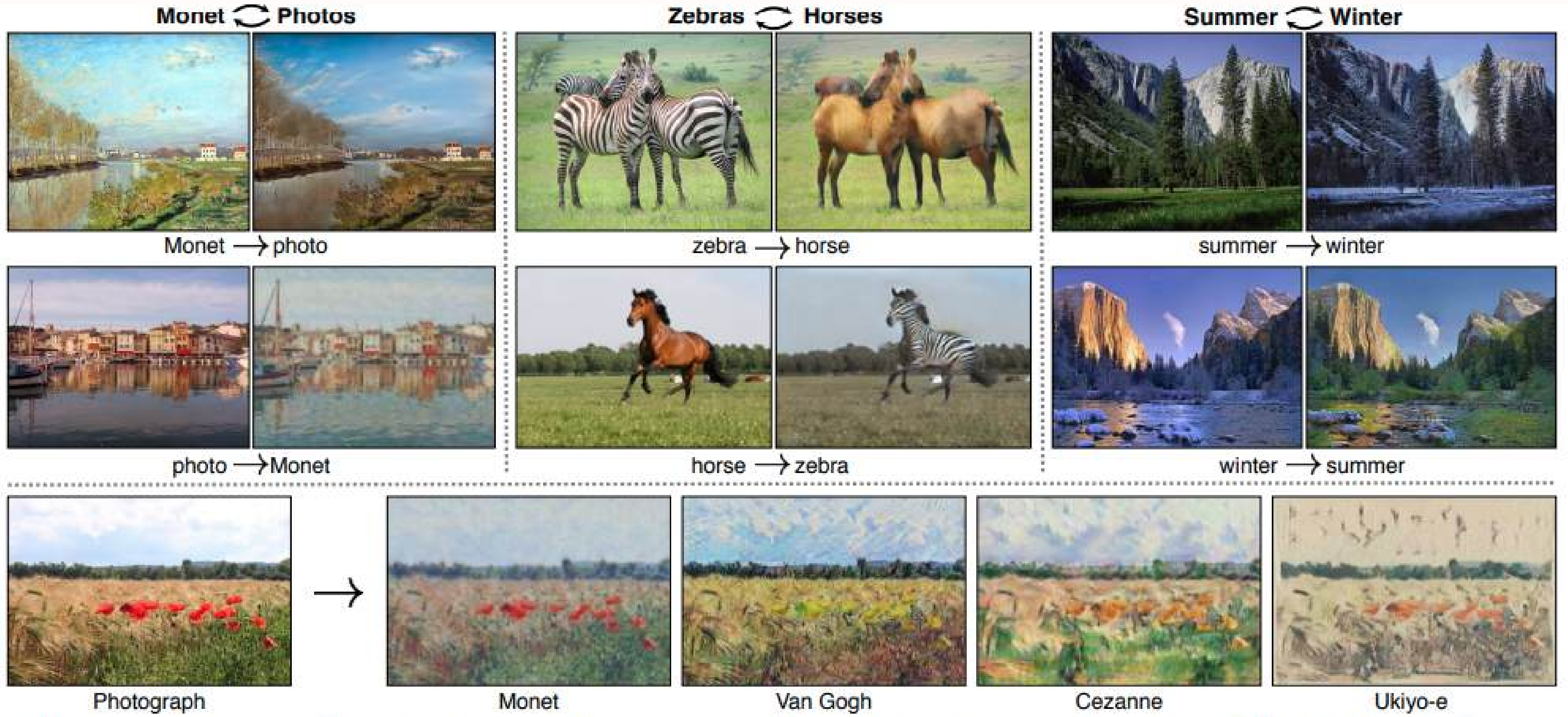
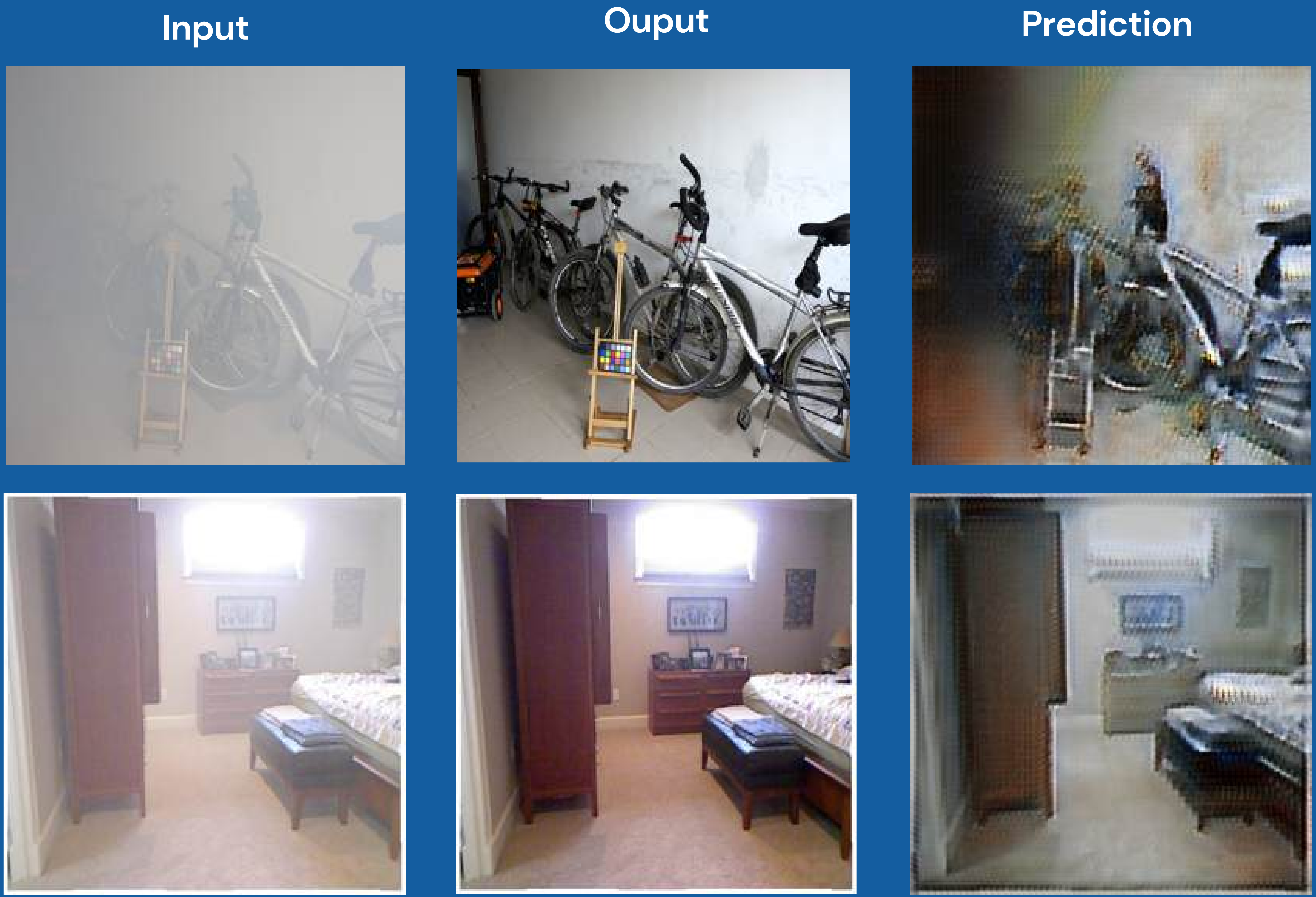


Figure 1: Given any two unordered image collections X and Y, our algorithm learns to automatically "translate" an image from one into the other and vice versa: *(left)* Monet paintings and landscape photos from Flickr; *(center)* zebras and horses from ImageNet; *(right)* summer and winter Yosemite photos from Flickr. Example application *(bottom)*: using a collection of paintings of famous artists, our method learns to render natural photographs into the respective styles.

02 CycleGAN

2.5 Result

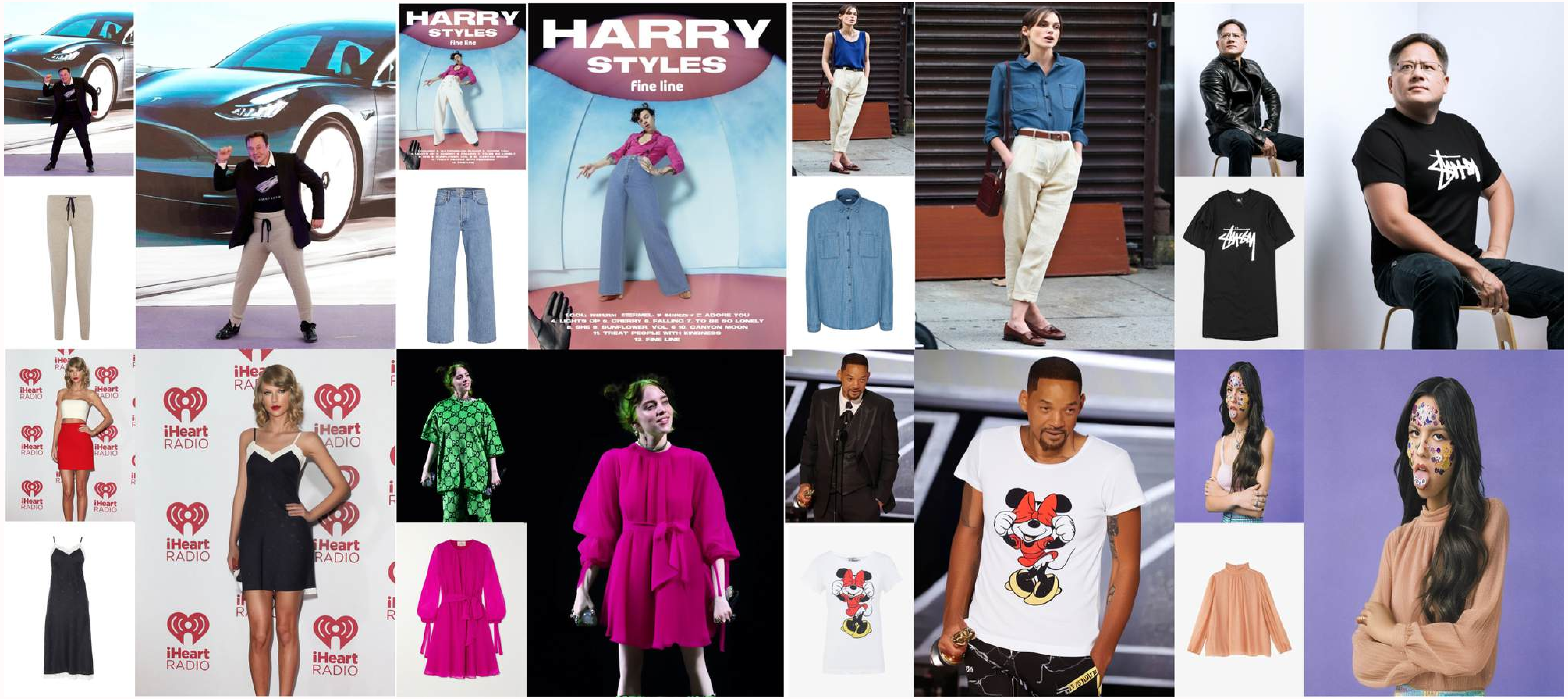


Input

Ouput

Prediction

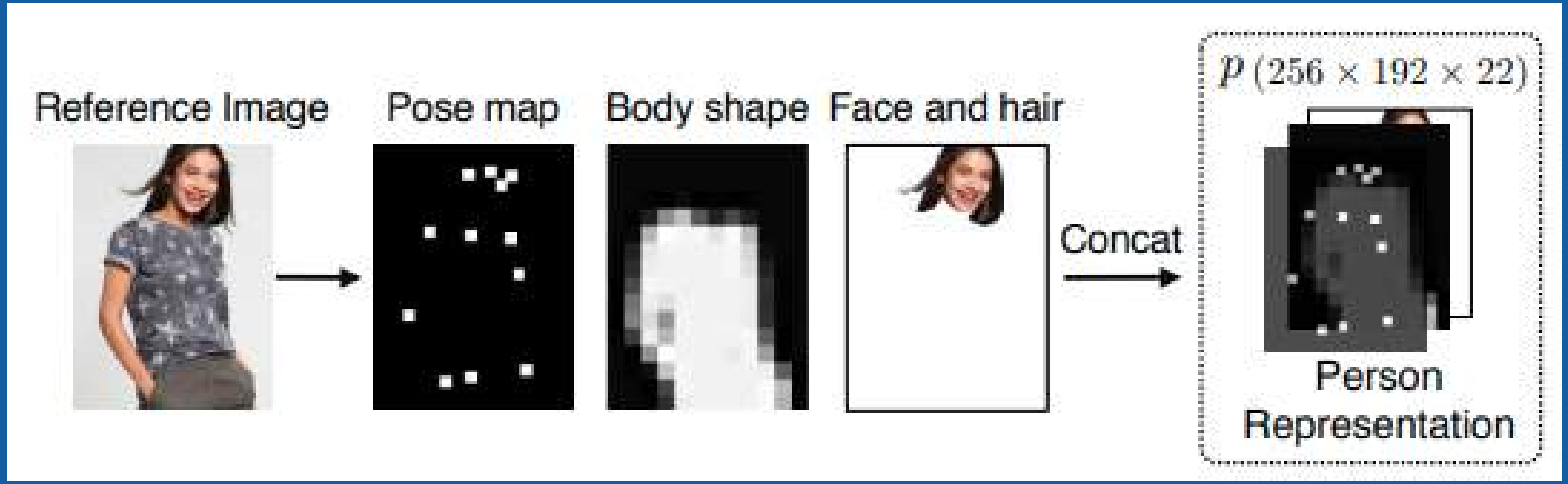
03 Virtual Try On

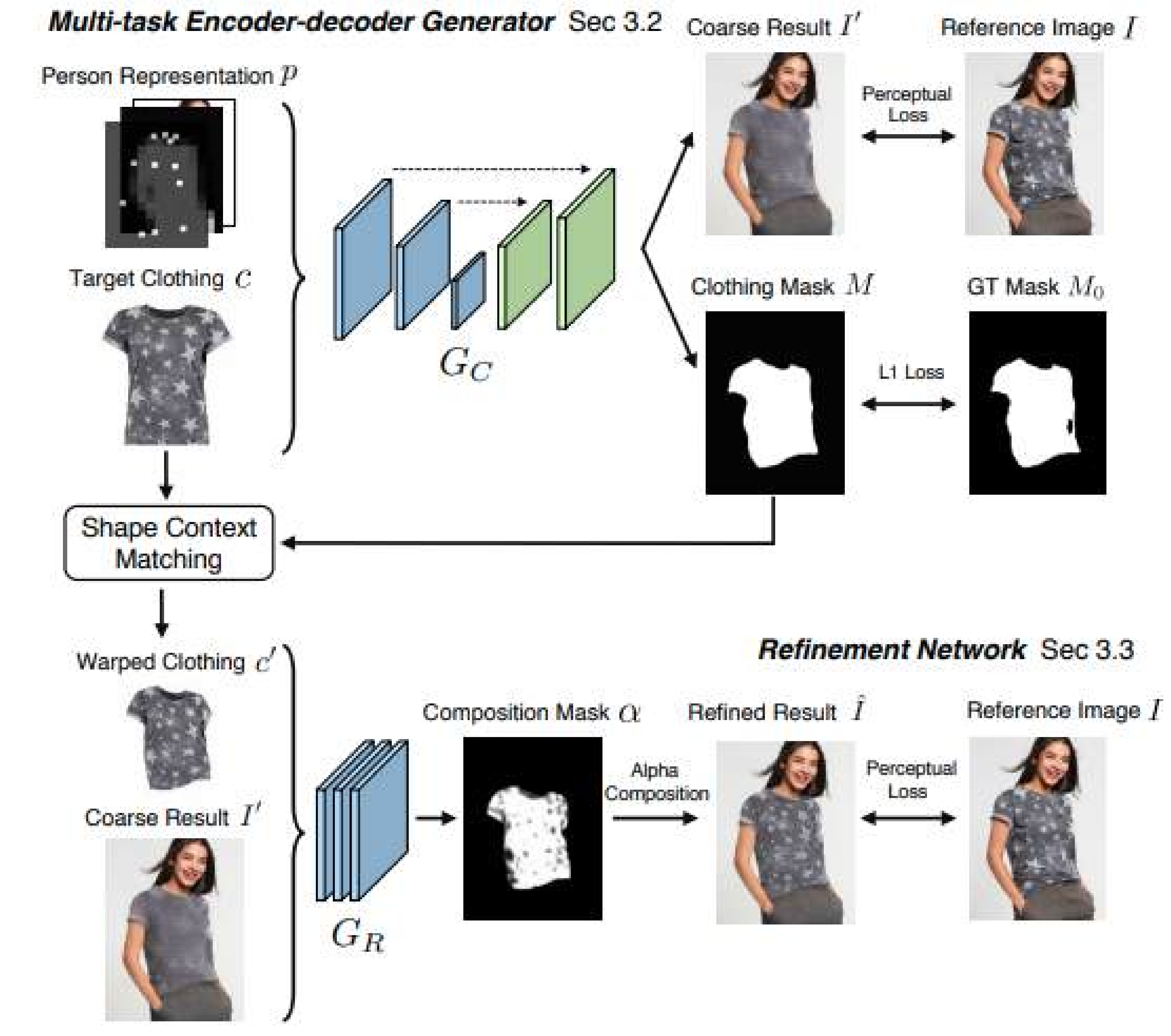


04

VITON

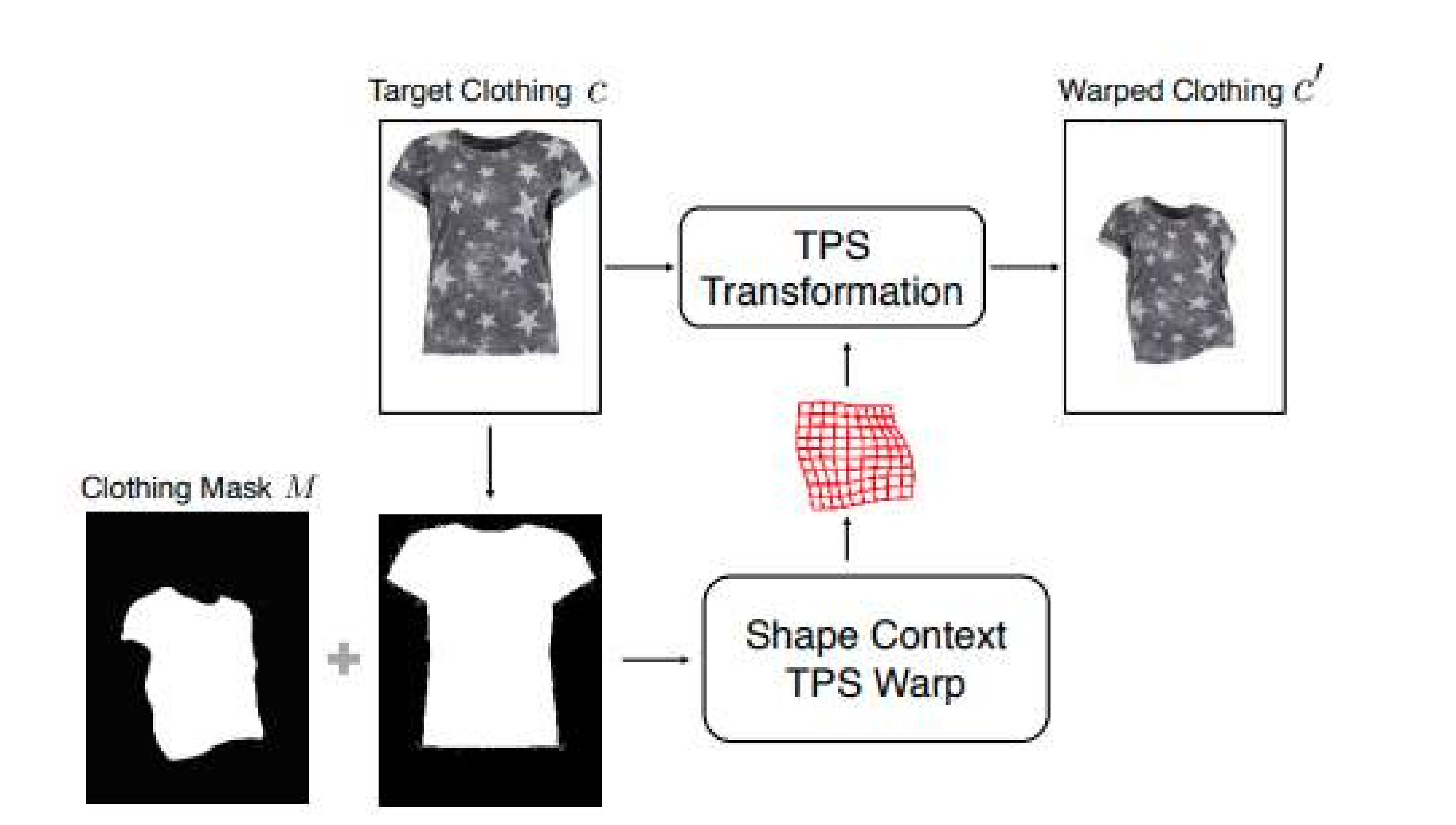
4.1 Architecture

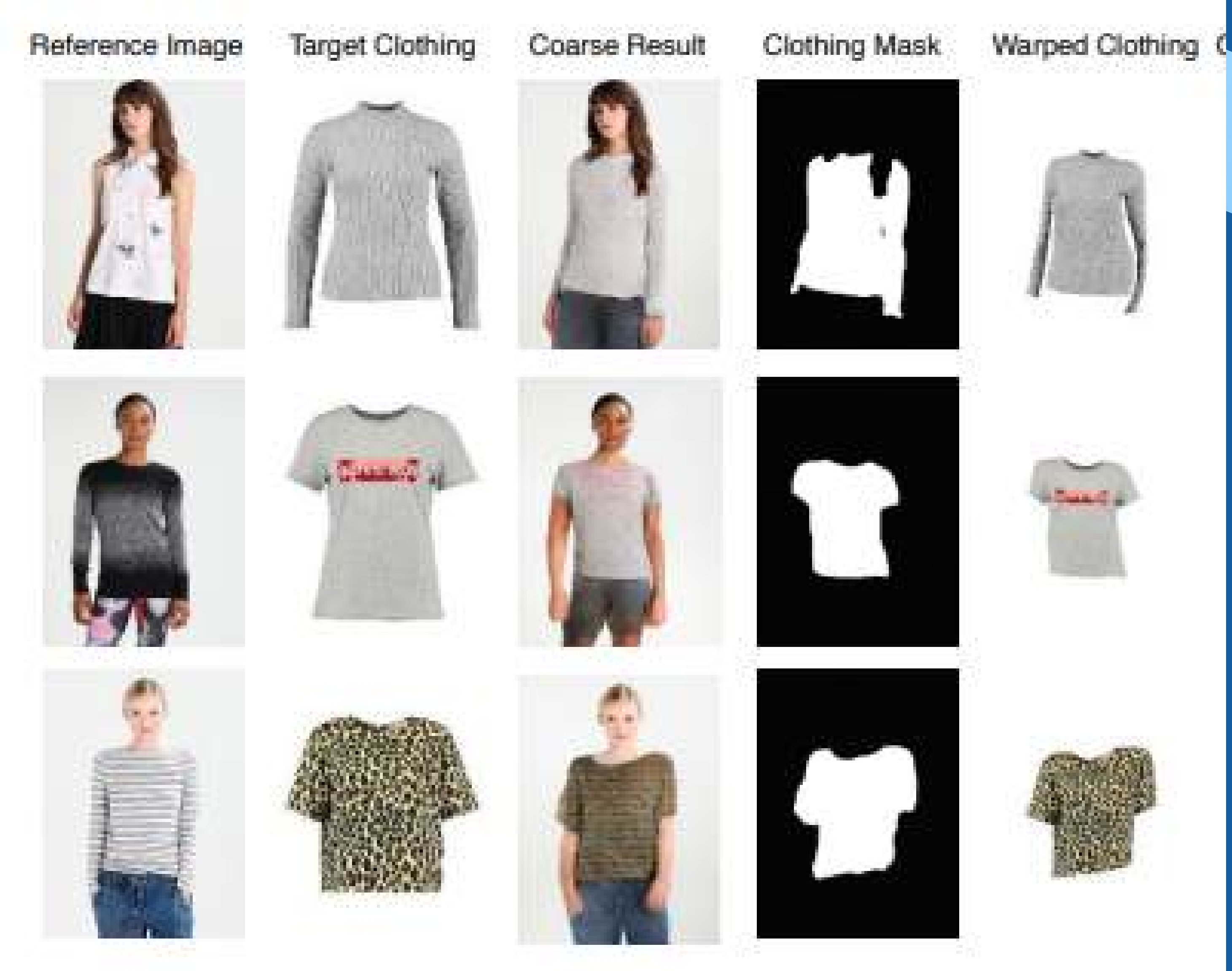




04 VITON

4.2 TPS Transformation



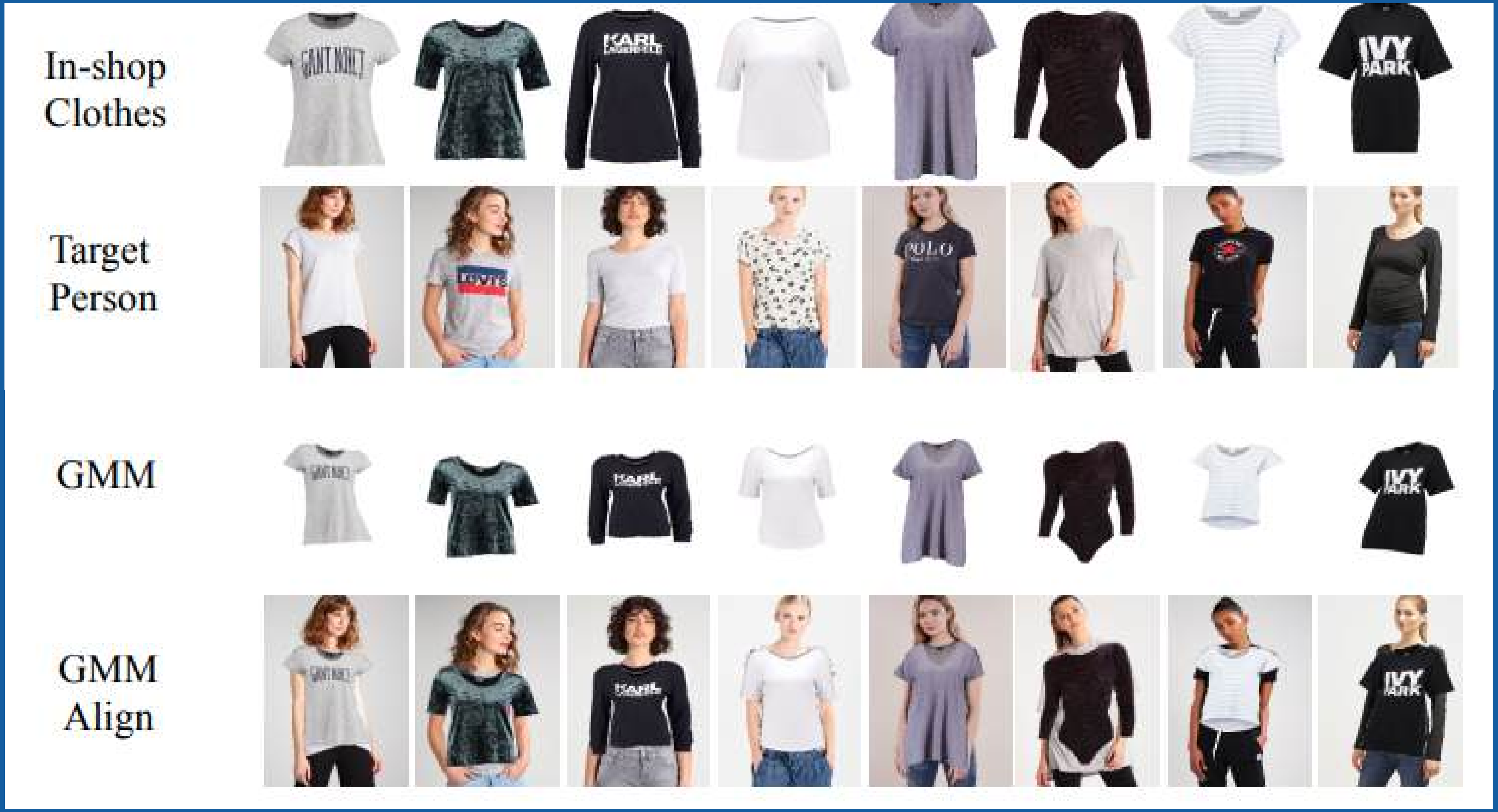


$$\sum_i \|f(P_i) - Q_i\|^2 + \lambda \iint \left(\frac{\partial^2 f}{\partial x^2} + 2\frac{\partial^2 f}{\partial x \partial y} + \frac{\partial^2 f}{\partial y^2}\right)^2 dx\, dy$$

$$I_o = M \odot \hat{c} + (1 - M) \odot I_r$$

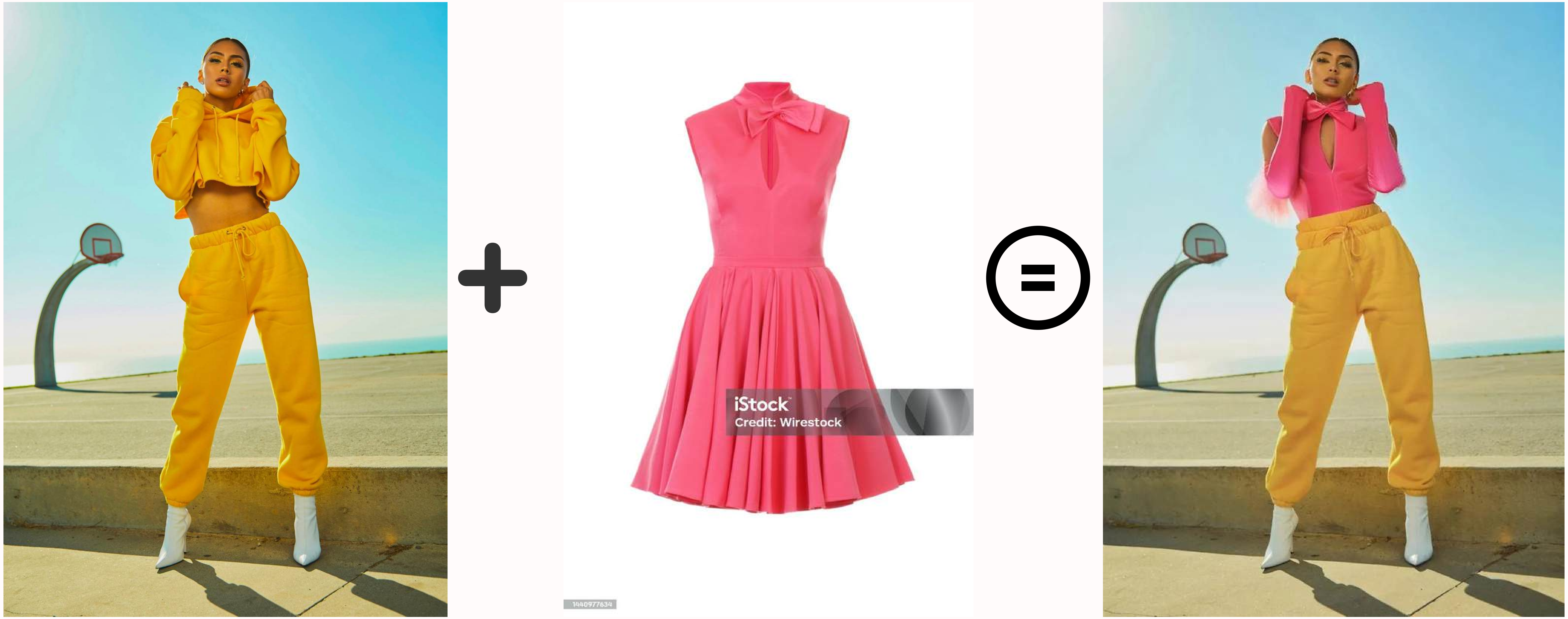
04 VITON

4.3 Result



04 VITON

4.5 Result



Kepco - UET

Trường Đại học Công nghệ

Thank's For Watching